EECE 5639 Computer Vision I

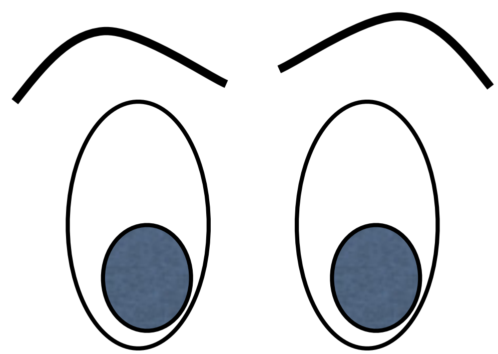
Lecture 15

Stereo: Epipolar Geometry

Next Class

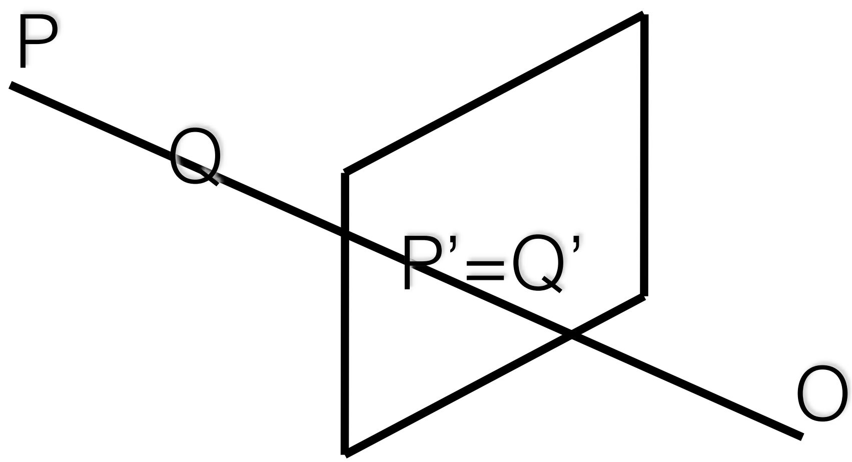
2nd midterm NEW DATE: FRIDAY 3/25

Stereo Vision



Why Stereo Vision?

2D images project 3D points into 2D:



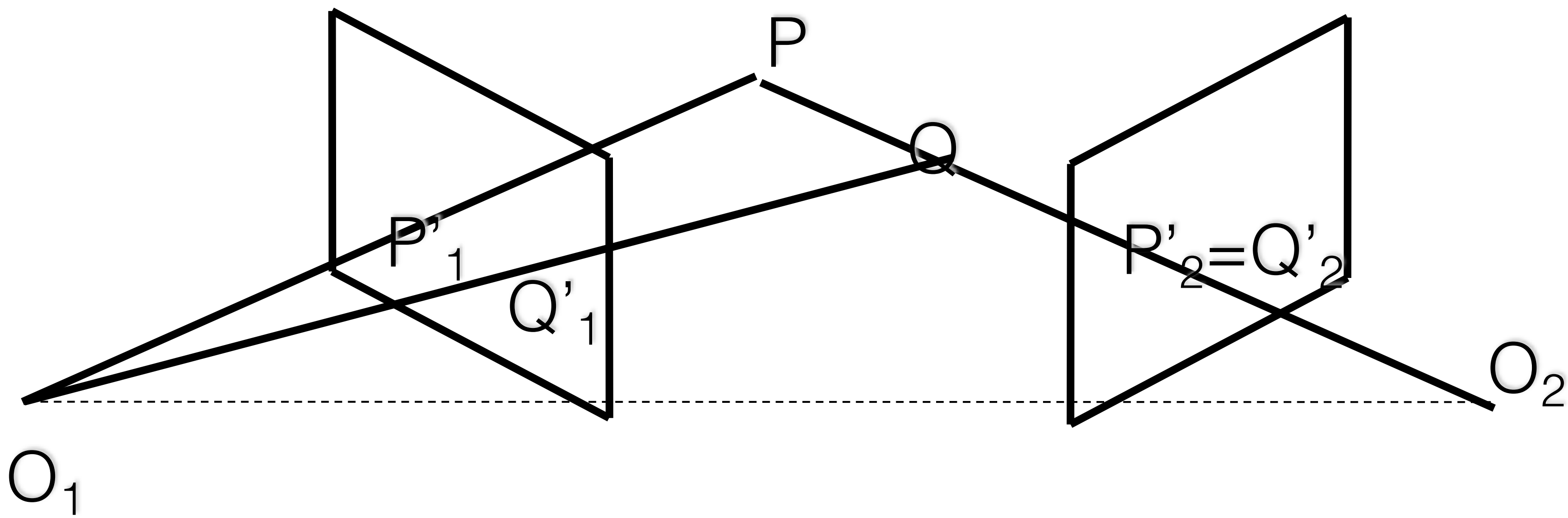
- 3D Points on the same viewing line have the same 2D image:
- 2D imaging results in depth information loss

Stereo Vision

Refers to the ability of:

The ability to infer information on the 3D structure and distance of a scene from two or more images taken from different viewpoints.

Recovering Depth Information:



Depth can be recovered with two images and triangulation.

Stereo Vision Problems:

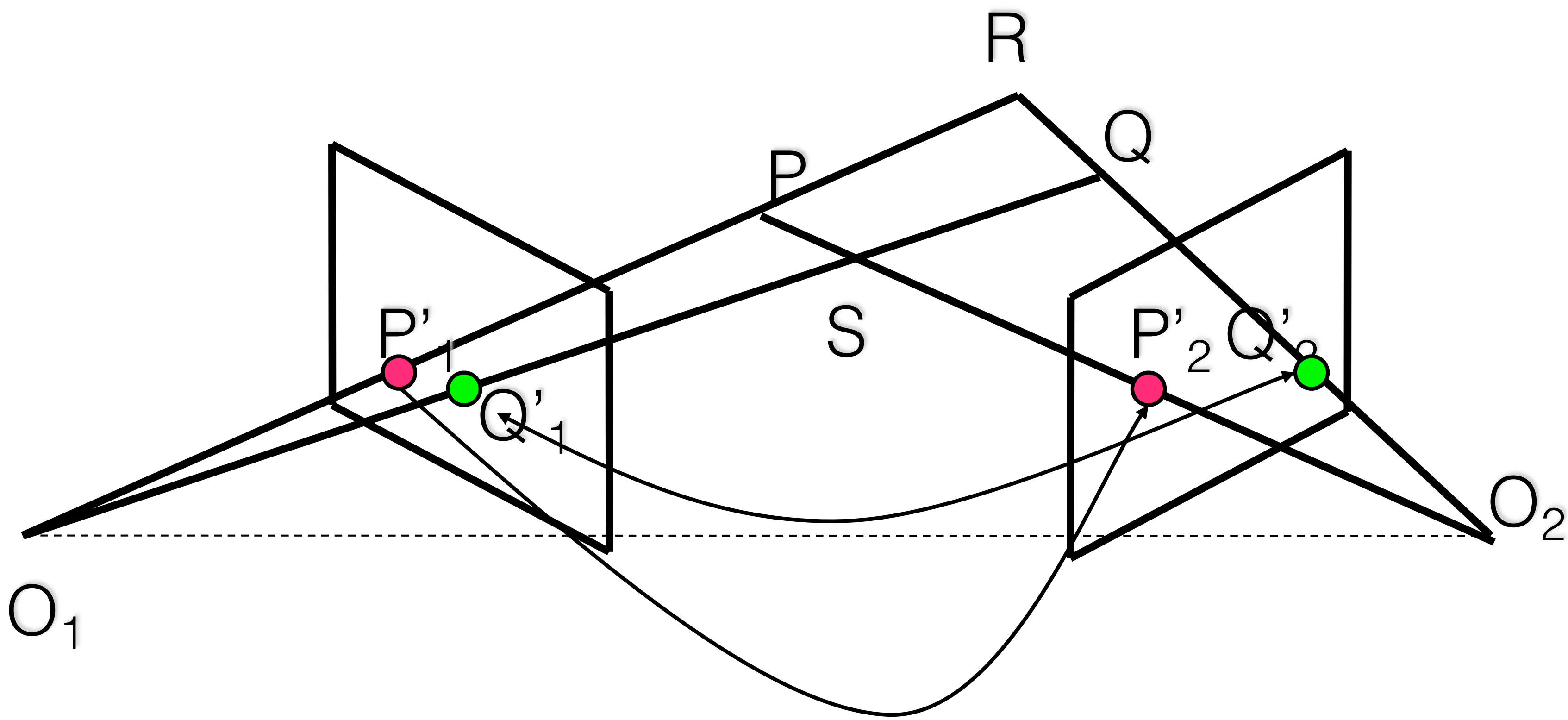
Correspondence Problem:

Determining which pixel on the left corresponds to which pixel on the right.

Reconstruction Problem:

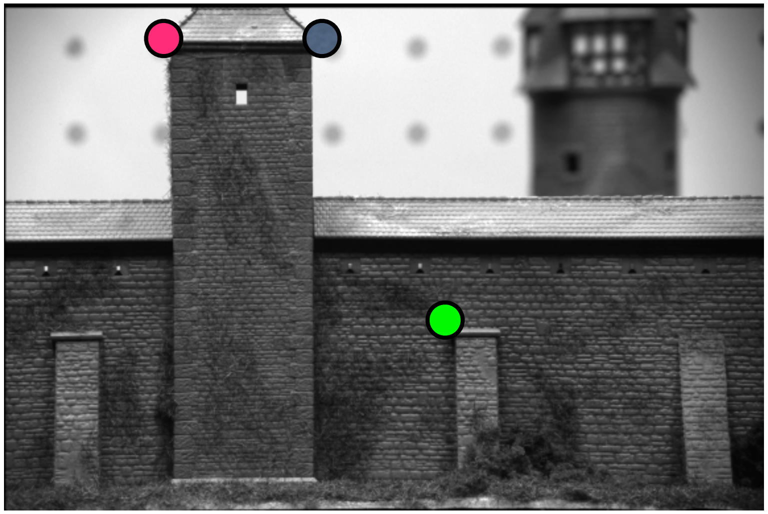
Given a number of correspondence pairs and camera geometry information, find location and 3D structure of the observed objects.

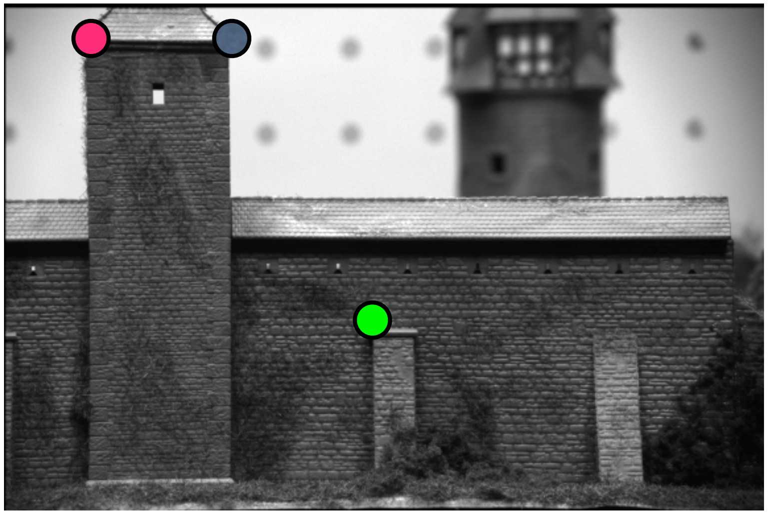
Finding Correspondences:



Wrong correspondences can result in large depth errors!

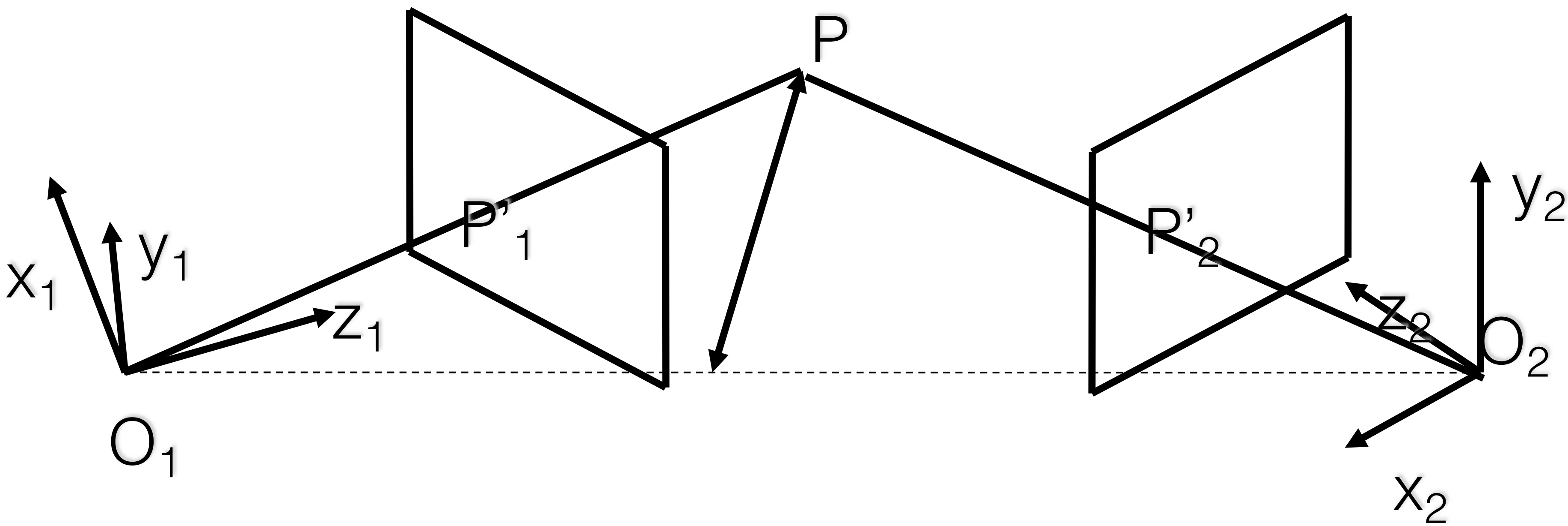
Finding Correspondences:





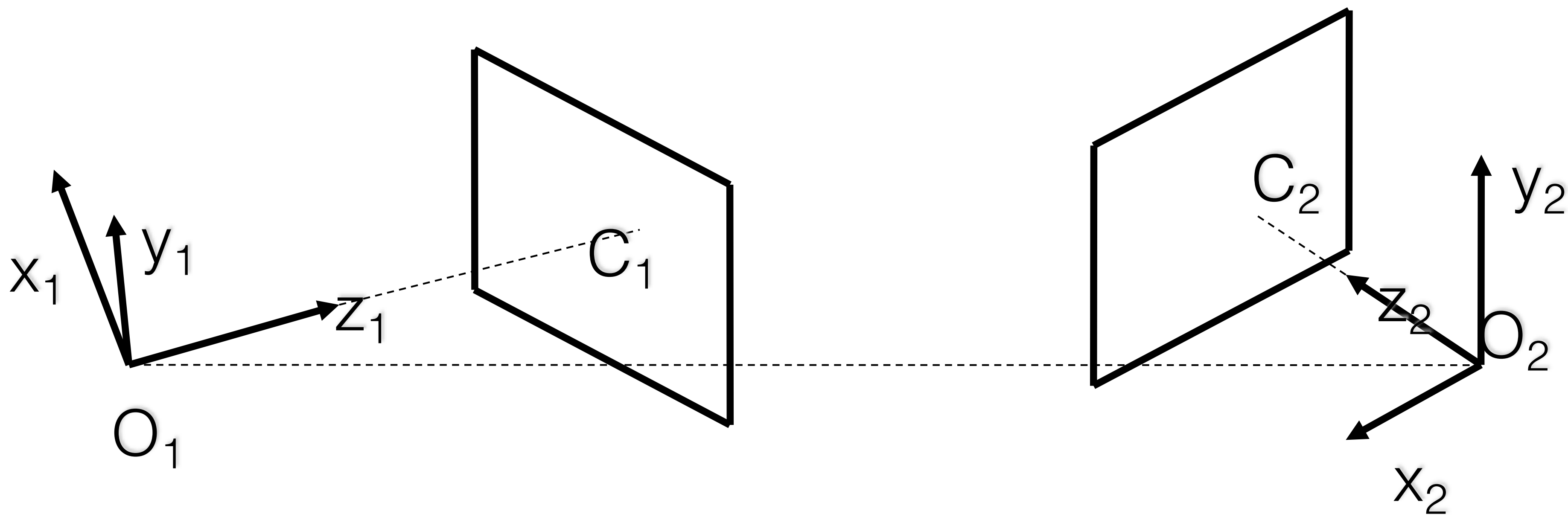
3D Reconstruction

Given the correspondences, “triangulate” to find depth:



The stereo rig must be calibrated!

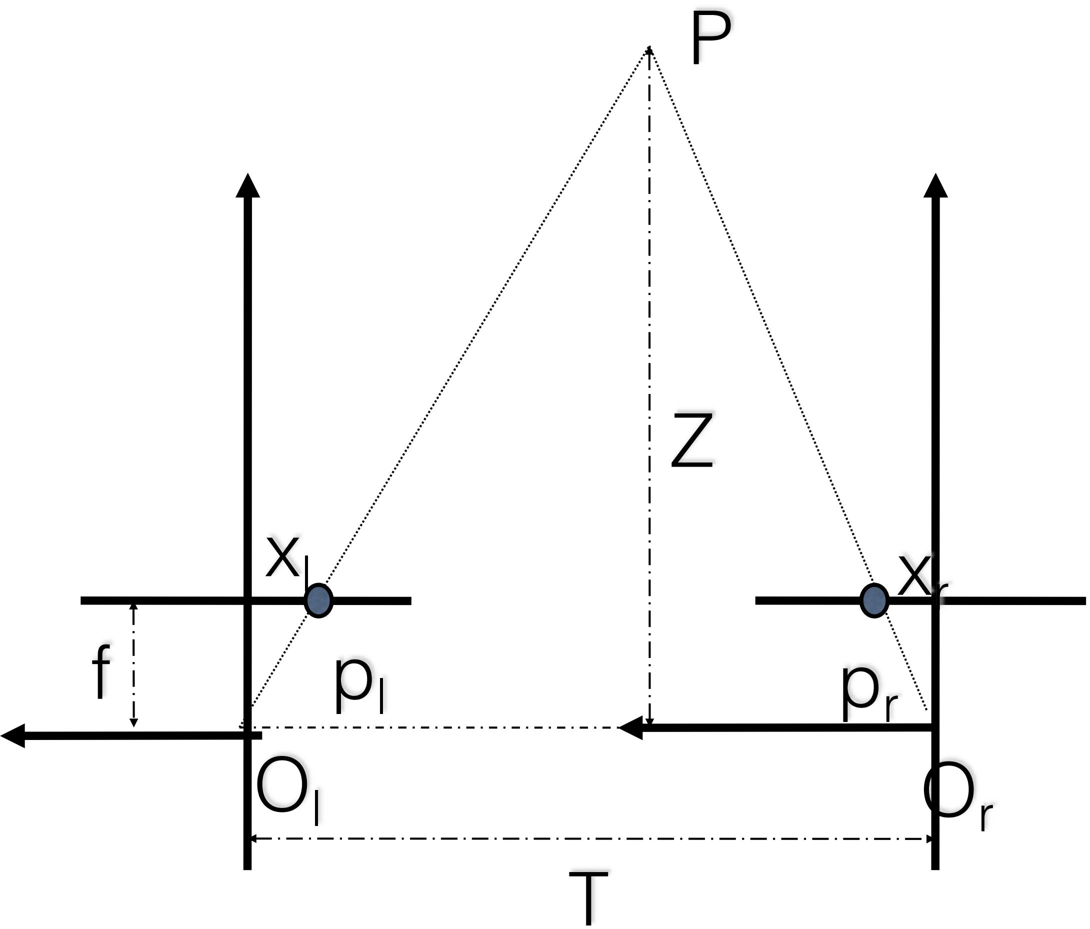
Parameters of a Stereo System



- Intrinsic:
  - f_1 and f_2: focal lengths
  - c_1 and c_2: principal points
  - Pixel size

- Extrinsic
  - Transformation (R,T) between cameras

A simple stereo system



$$\frac{T + x_l - x_r}{Z - f} = \frac{T}{Z}$$

$$Z = f\frac{T}{x_r - x_l}$$

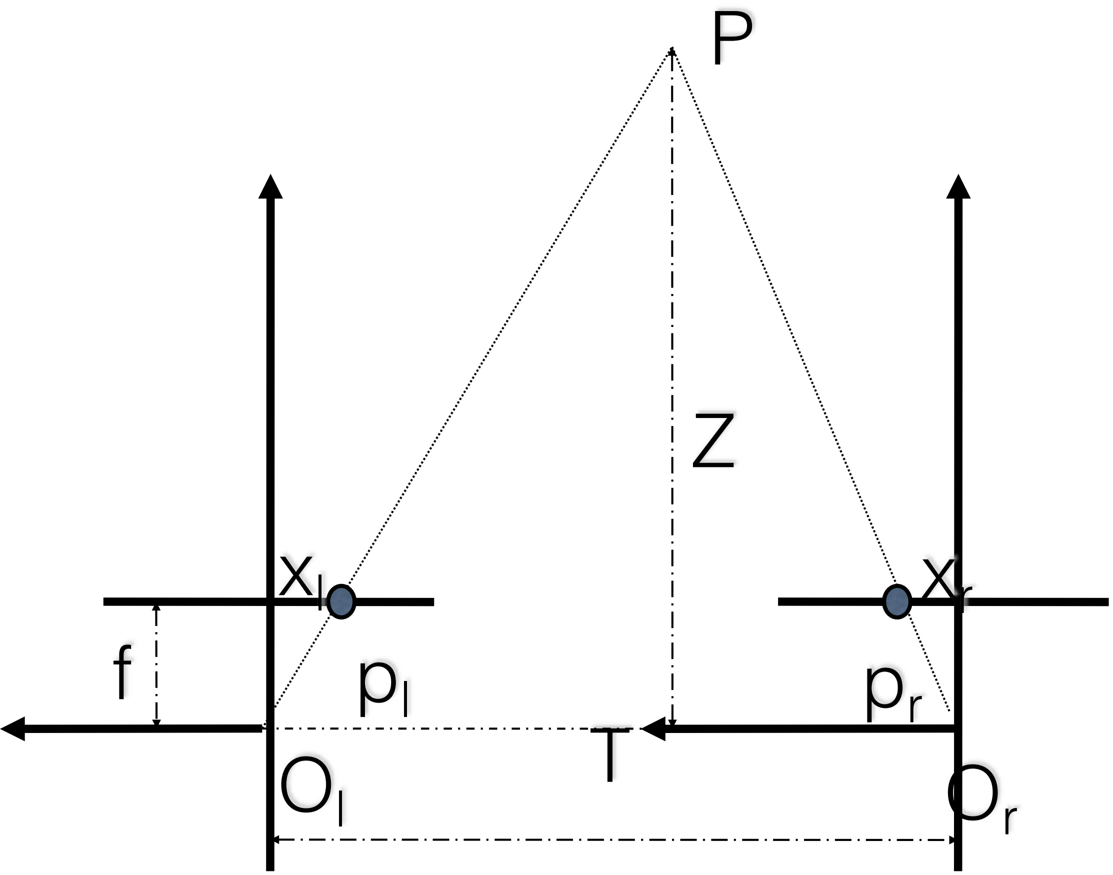
$$\boxed{Z = f\frac{T}{d}}$$

Disparity: $d = x_r - x_l$

T is the stereo baseline

d measures the difference in retinal position between corresponding points

A simple stereo system



$$\boxed{Z = f\frac{T}{d}}$$

- Depth is inversely proportional to disparity

The Correspondence Problem

Basic assumptions:

Most scene points are visible in both images

Corresponding image regions are similar

These assumptions hold if:

The distance of the fixation point from the cameras is much larger than the stereo baseline: $Z >> T$

The Correspondence Problem

Is a "search" problem:

- Given an element in the left image, search for the corresponding element in the right image.

We must choose:

- Elements to match
- A similarity measure to compare elements

Correspondence Problem

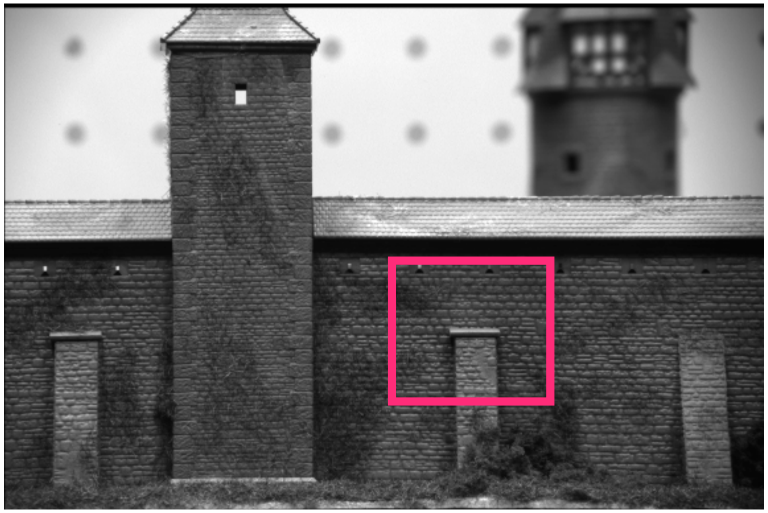
Two classes of algorithms:

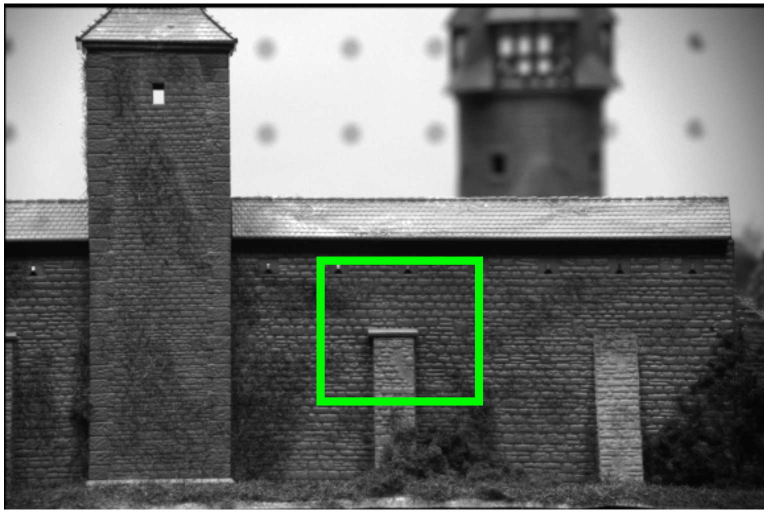
- Correlation-based algorithms
  - Produce a DENSE set of correspondences
- Feature-based algorithms
  - Produce a SPARSE set of correspondences

Correlation-based Algorithms

Elements to be matched:

image WINDOWS of fixed size.





Finding the disparity map

Inputs:

- Left image I_l
- Right image I_r

Parameters that must be chosen:

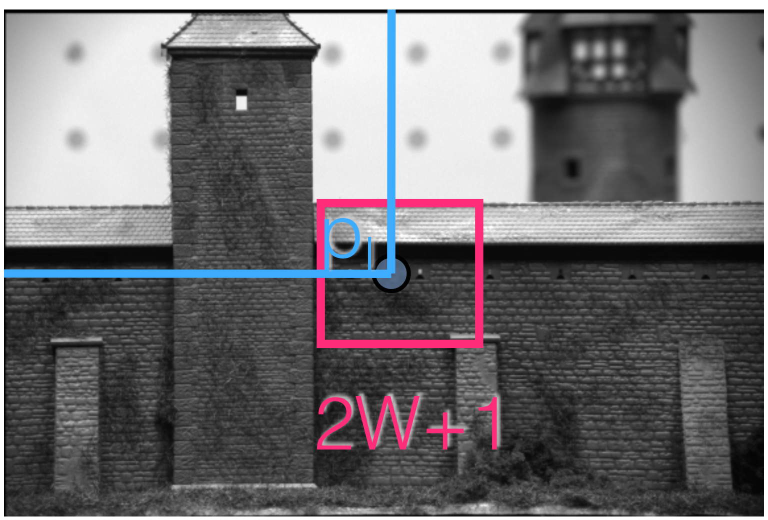
- Correlation Window size 2W+1
- Search Window size ω
- Similarity measure Ψ

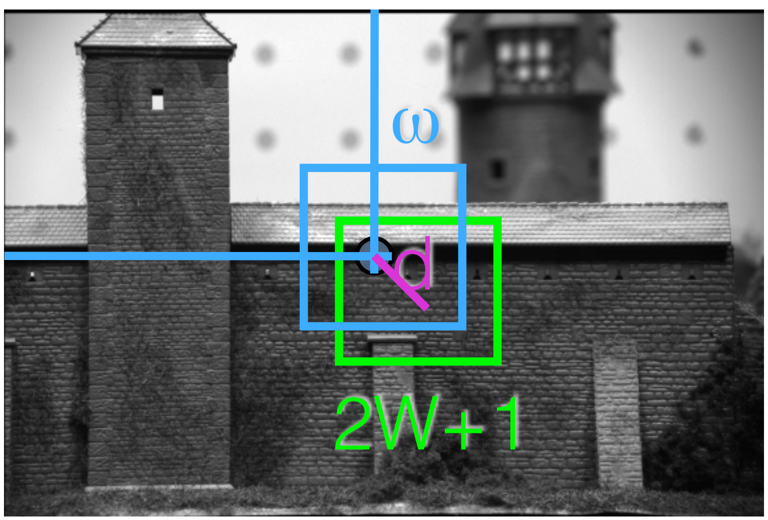
CORR_MATCHING Algorithm

Let p_l and p_r be pixels on the I_l and I_r

Let $R(p_l)$ be the search window ω x ω on I_r associated with p_l

Let d be the displacement between p_l and a point in $R(p_l)$.





CORR_MATCHING Algorithm

For each pixel $p_l=[i,j]$ in I_l do:

For each displacement $d=[d_1,d_2]$ in $R(p_l)$ compute:

$$C(d) = \sum_{l=-W}^{l=W} \sum_{k=-W}^{k=W} \Psi(I_l(i+k,j+l), I_r(i+k-d_1, j+l-d_2))$$

The disparity at p_l is the vector d with best C(d) over $R(p_l)$

(i.e. max. C_{fg}, or min. SSD)

Output the disparity for each pixel p_l

How do we set W, R and ω ?

W (width of the correlation window):

should be based on the “scale” of the scene.

$R(P_l)$ and ω (search window):

Should be estimated based on the range of scene distances and the baseline:

$Z = fT/d$ or $d = fT/Z$

Feature-based Methods

Similar to Correlation-based methods, but:

- They only search for correspondences of a **sparse** set of image features.
- Correspondences are given by the most similar feature pairs.
- Similarity measure must be adapted to the type of feature used.

Feature-based Methods:

Features most commonly used:

- Corners, SIFT
  - Similarity measured in terms of:
    - surrounding gray values (SSD, Cross-correlation)
    - SIFT descriptor
    - location
- Edges, Lines
  - Similarity measured in terms of:
    - orientation
    - contrast
    - coordinates of edge or line’s midpoint
    - length of line

Example: Comparing lines

l_l and l_r: line lengths

θ_l and θ_r: line orientations

(x_l,y_l) and (x_r,y_r): midpoints

c_l and c_r: average contrast along lines

ω_l ω_θ ω_m ω_c : weights controlling influence

$$S = \frac{1}{\omega_l(l_l - l_r)^2 + \omega_\theta(\theta_l - \theta_r)^2 + \omega_m[(x_l - x_r)^2 + (y_l - y_r)^2] + \omega_c(c_l - c_r)^2}$$

The more similar the lines, the larger S is!

FEATURE_MATCHING Algorithm

Inputs:

- I_l and I_r
- Set of features on the left and right

Things that must be chosen:

- Search Window
- Similarity measure

FEATURE_MATCHING Algorithm

For each feature f_l in the left image:

Compute the similarity measure between f_l and every feature in the search window $R(f_l)$

Select the feature in $R(f_l)$ that maximizes the similarity measure.

Save the correspondence and the disparity of f_l

Output the list of correspondences and disparities.

Which method should we use?

Correlation methods:

dense maps, good for surface reconstruction

Require textured images

Sensitive to illumination variations

Inadequate for very different viewpoints

Feature methods:

Sparse maps, good for navigation

Require prior knowledge of type of scene

Must find features first

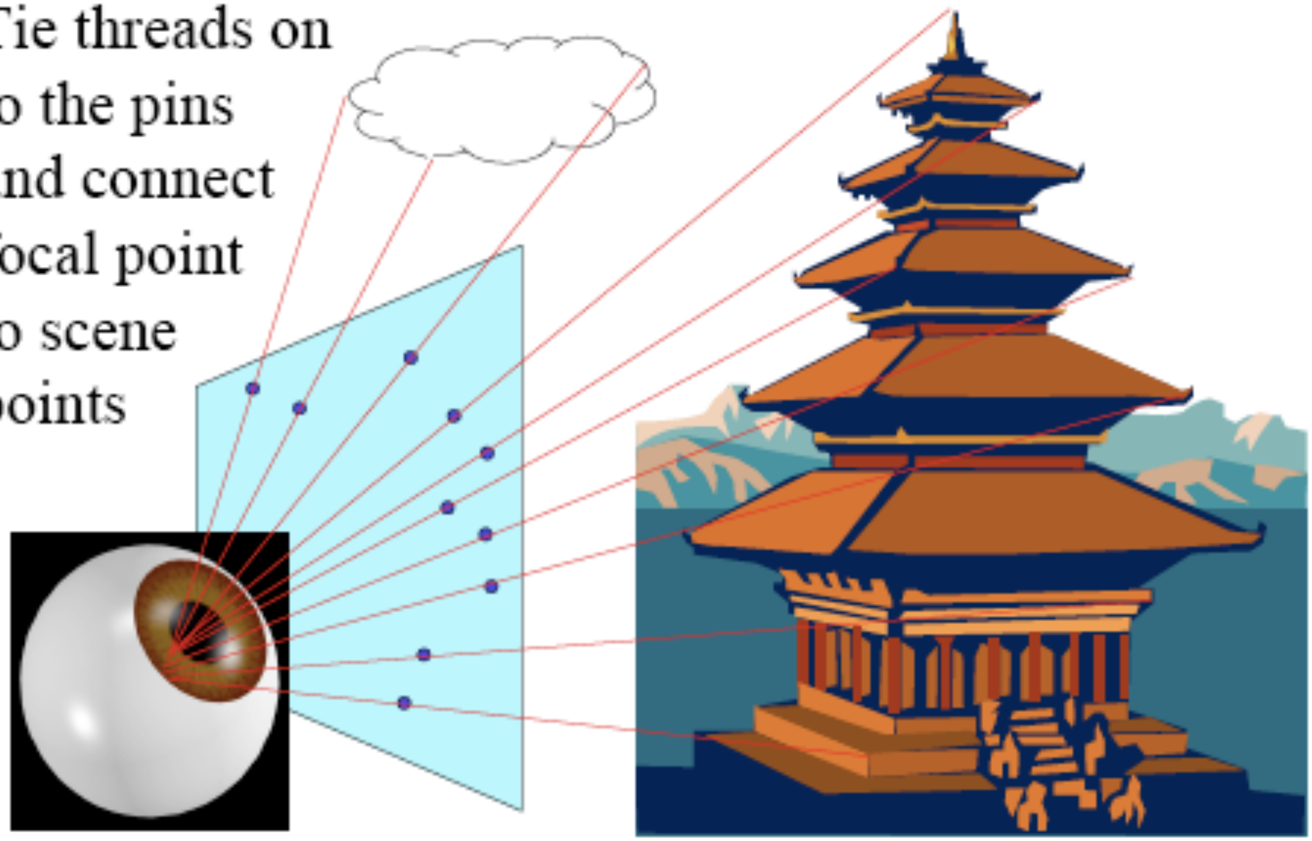
Constraining the Search Space

Finding correspondences is a search problem.

Geometry can be used to constrain the search.

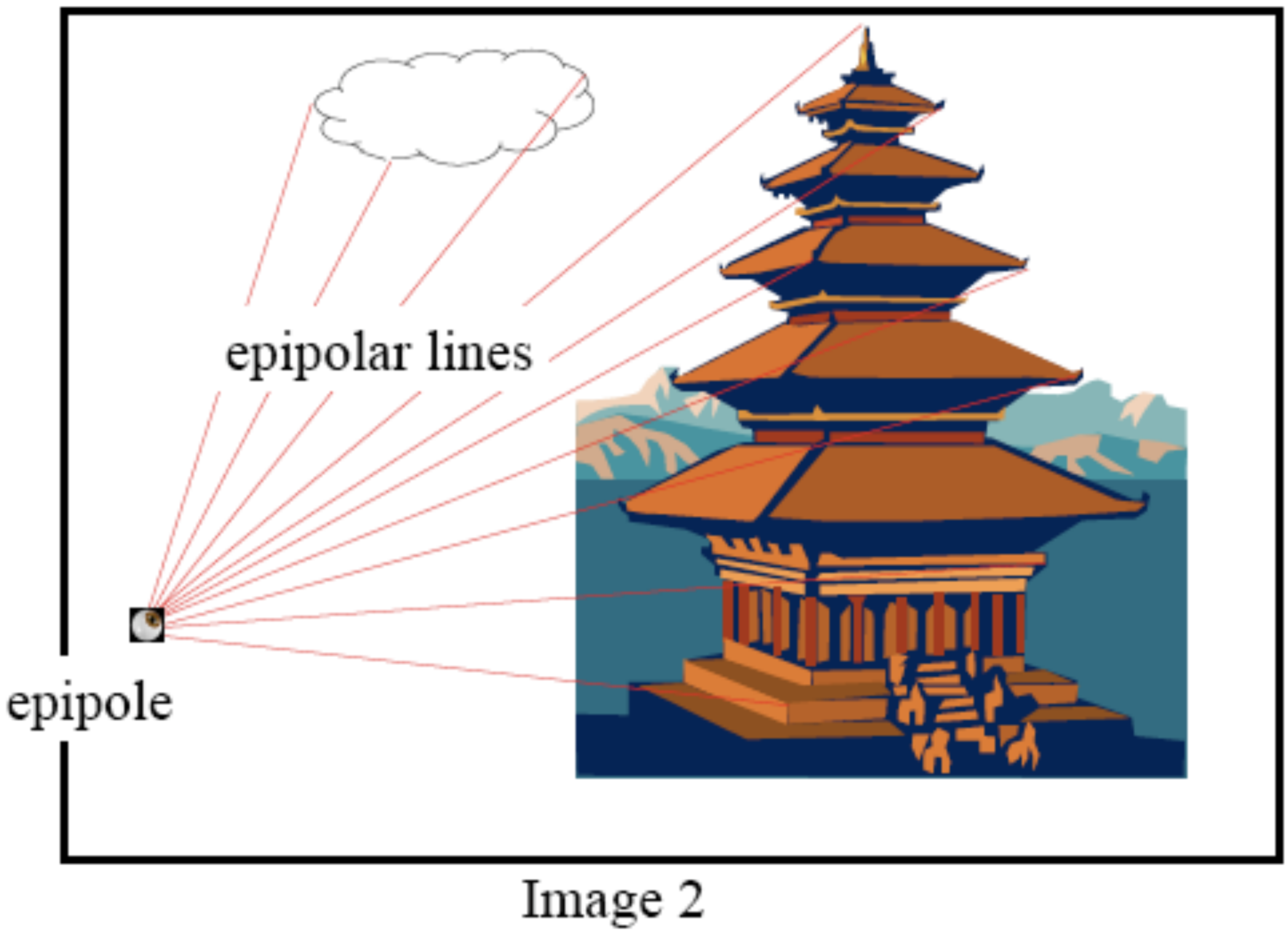
Rays to Pts in the Scene

Tie threads on
to the pins
and connect
focal point
to scene
points



Now what would this look like to a second observer?

Rays Seen from Second Observer



Rays Seen by the First Viewer

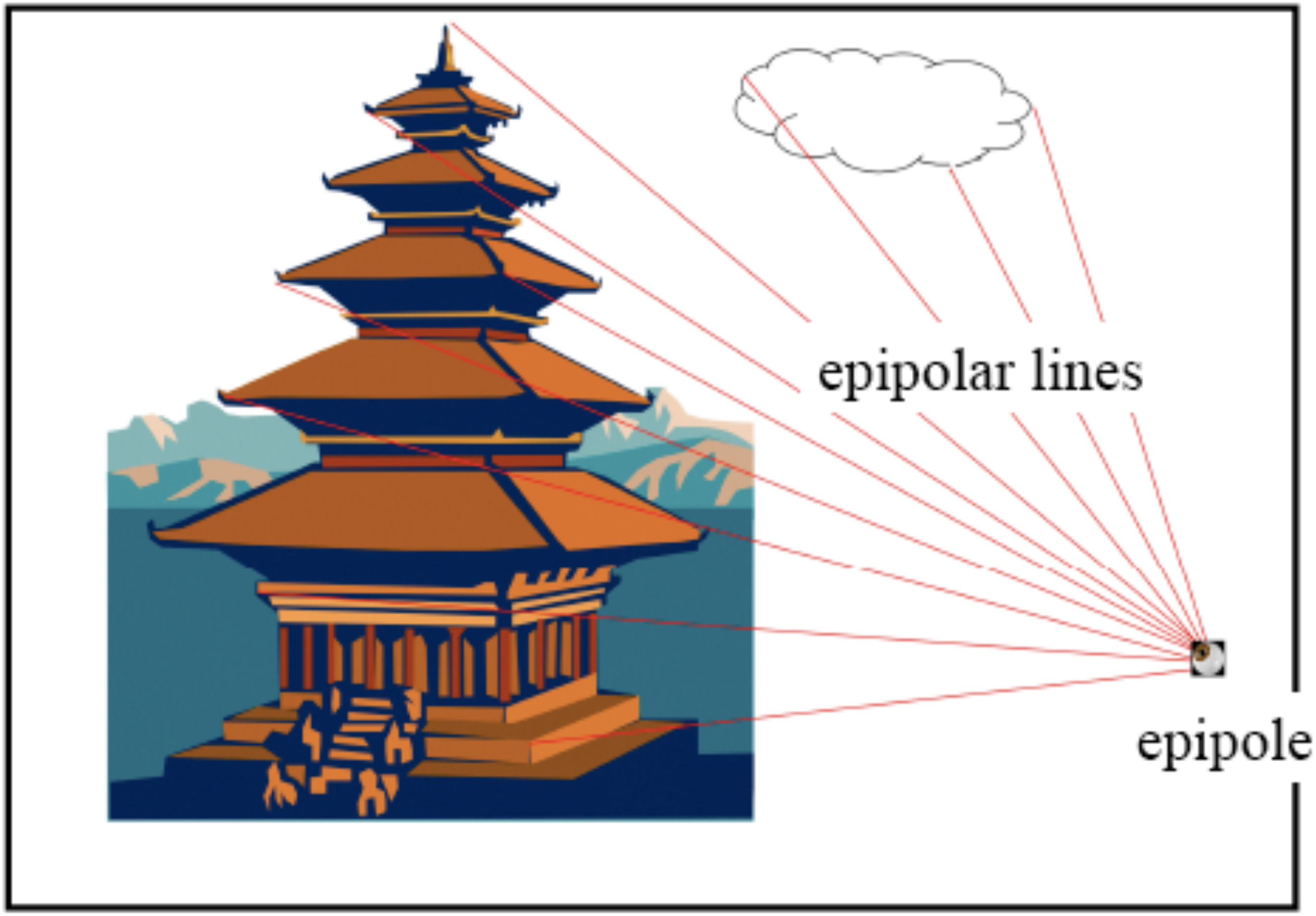
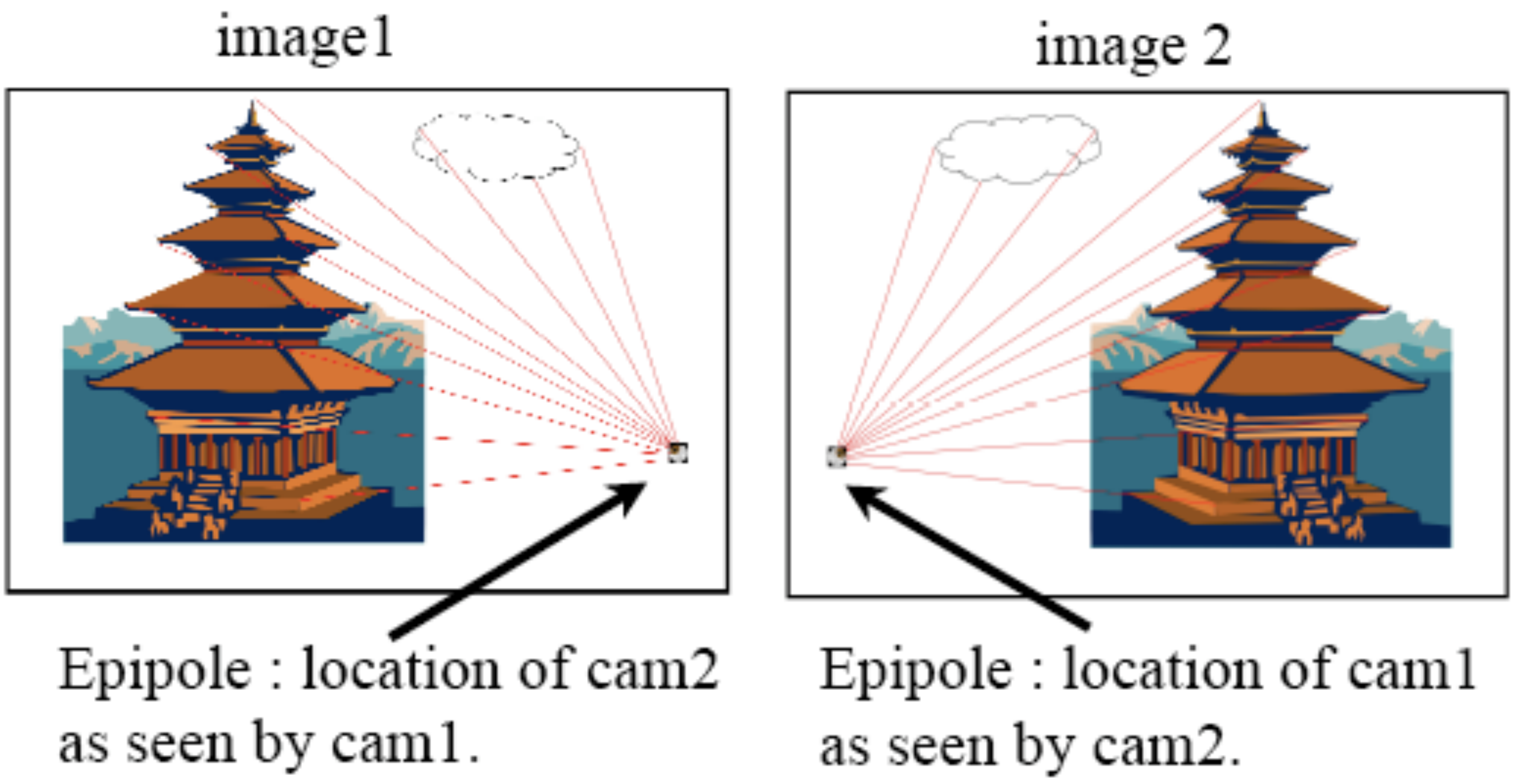
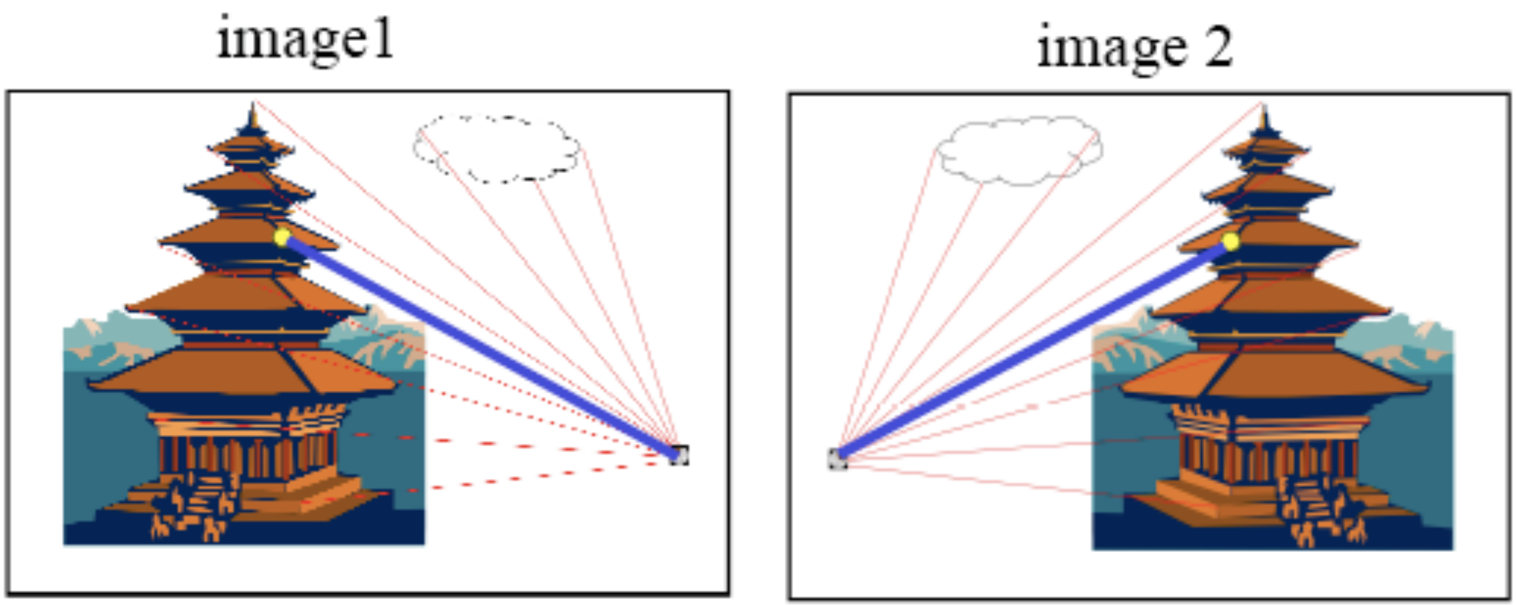


Image 1

Epipolar Geometry

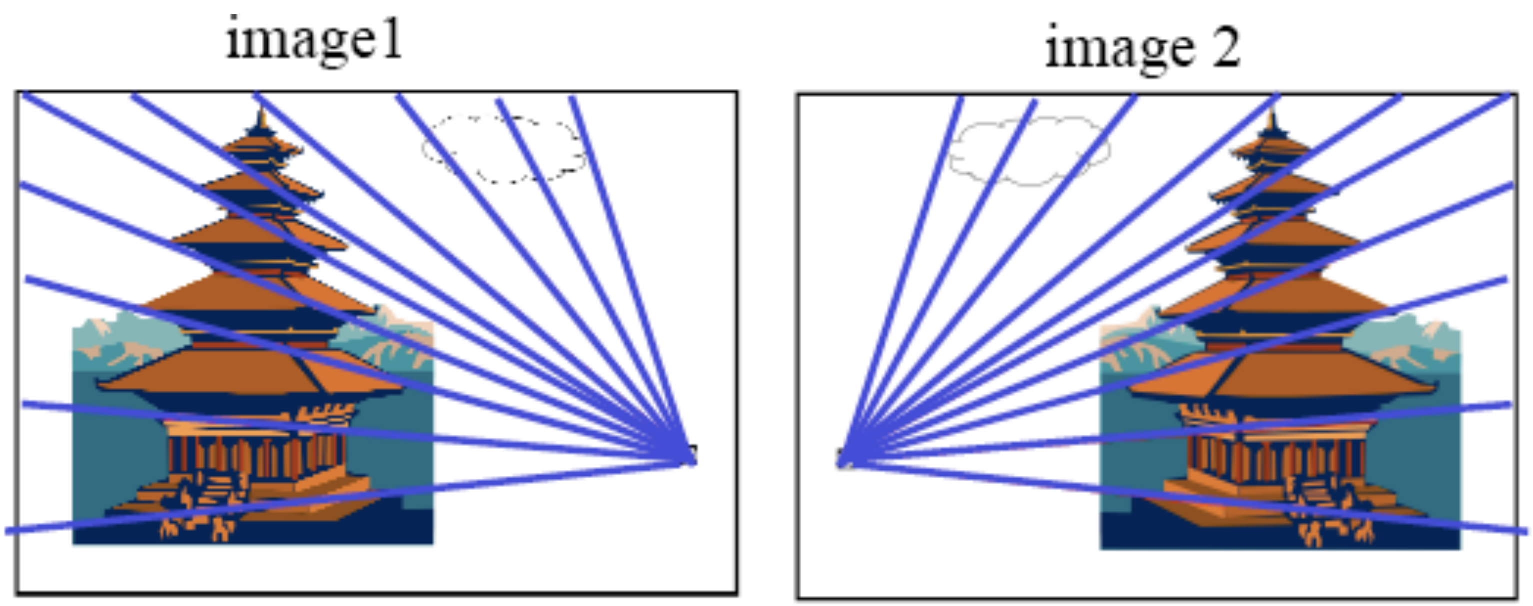


Epipolar Geometry



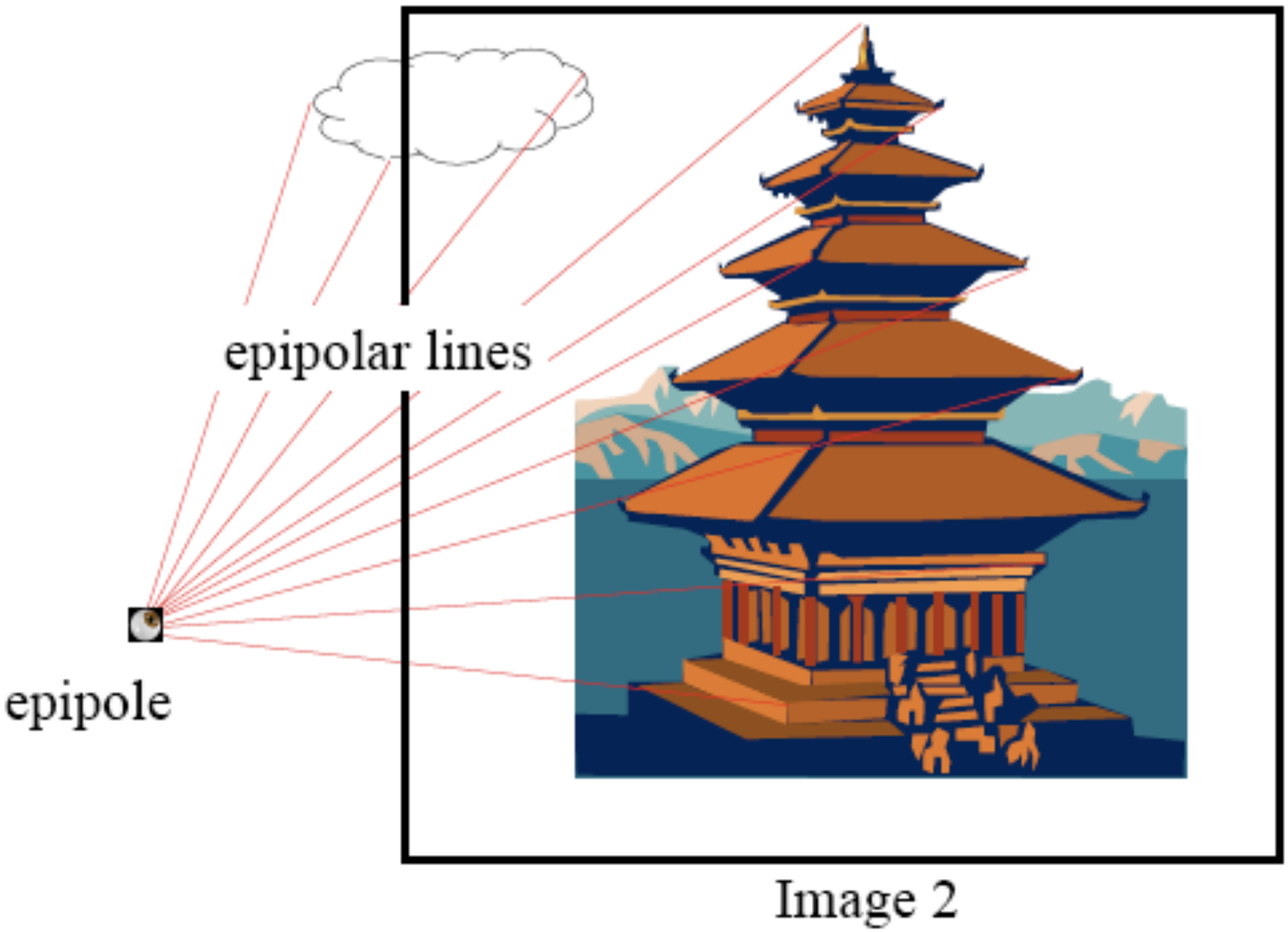
Corresponding points
lie on conjugate epipolar lines

Epipolar Geometry

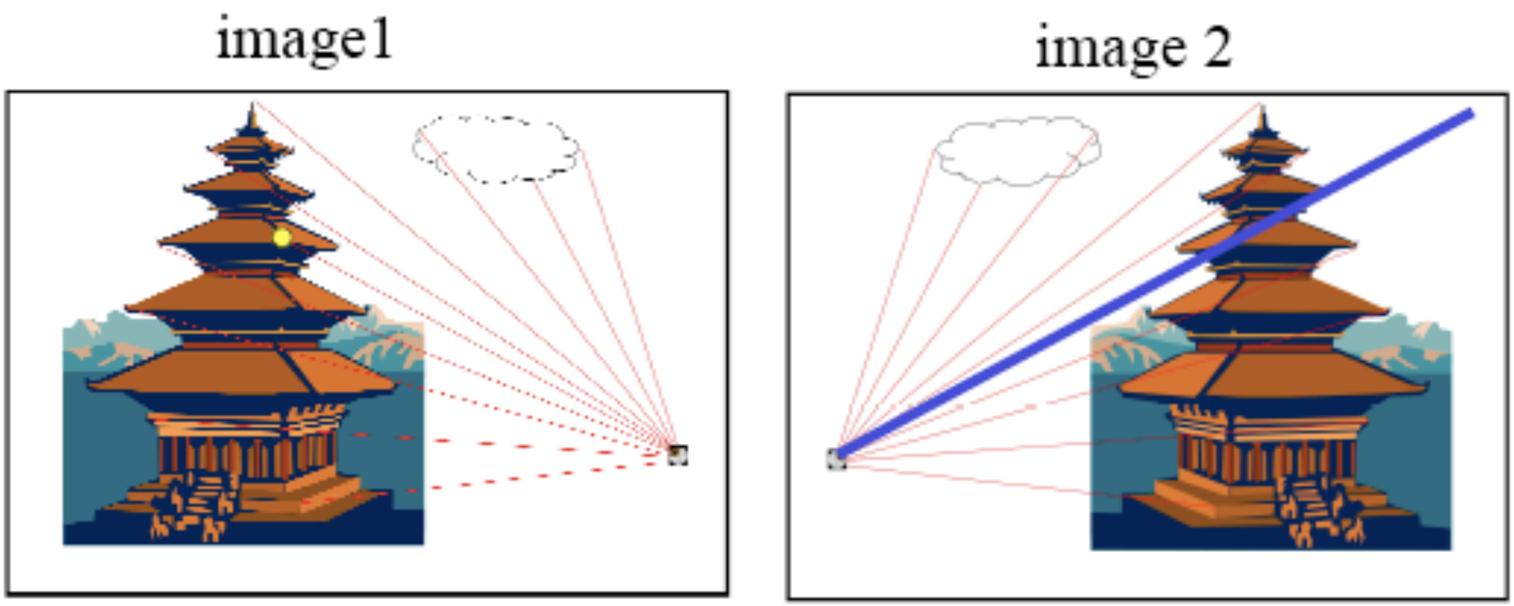


Conjugate epipolar lines induce a generalized 1D "scan-line" ordering on the images (analogous to traditional scan line ordering of rows in an image)

Epipole, not necessarily in the image



How do we find Epipolar Lines?



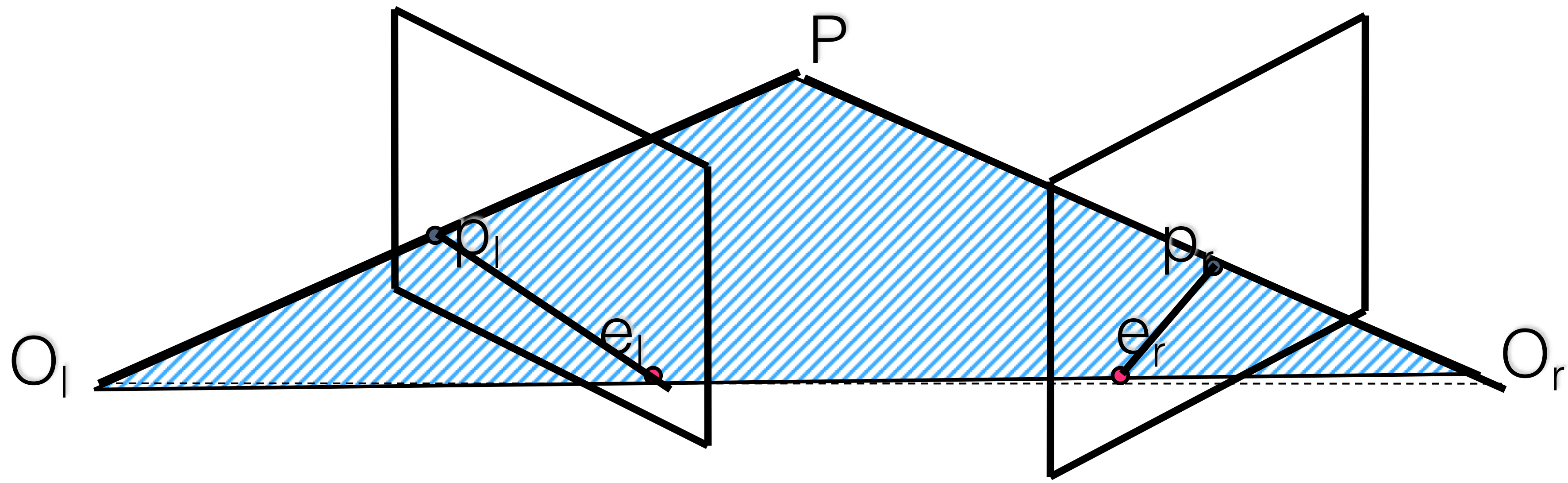
Given a point in one image, how do we determine the corresponding epipolar line to search along in the second image?

Essential Matrix

The ESSENTIAL matrix is a 3x3 matrix that "encodes" the epipolar geometry of two views.

Given a point in an image, multiplying by the Essential Matrix, will tell us the EPIPOLAR line in the second image where the corresponding point must be.

Epipolar Geometry



Epipoles:

- e_l: left image of O_r
- e_r: right image of O_l

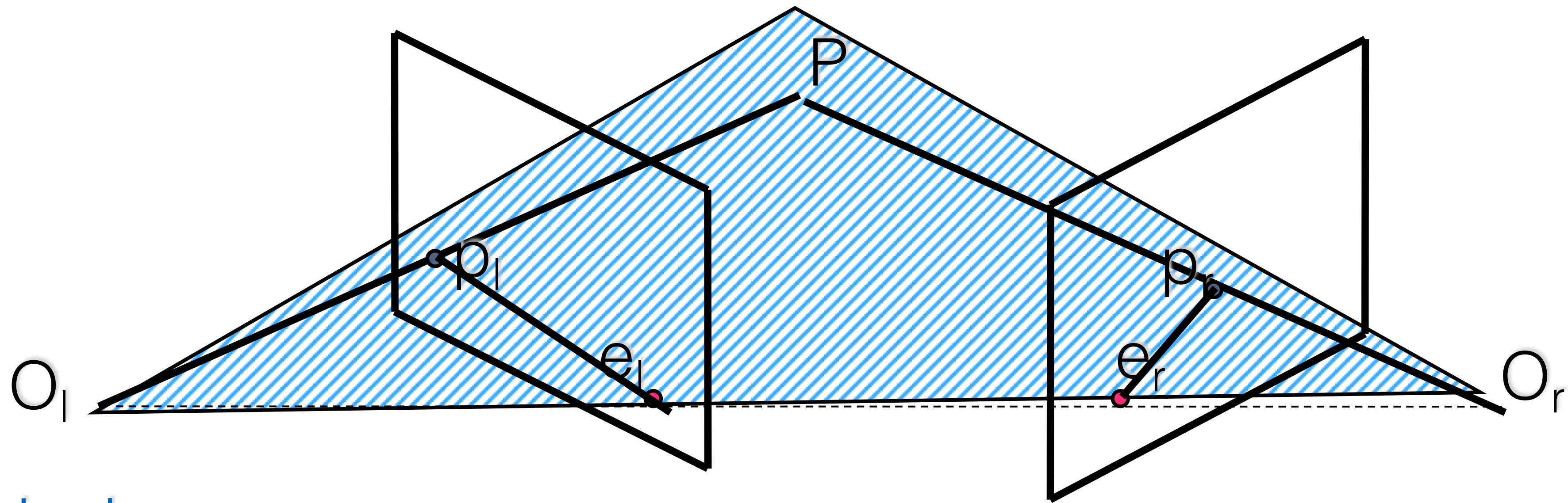
Epipolar plane:

- Three points: O_l,O_r, and P define an epipolar plane

Epipolar lines and epipolar constraint:

- Intersections of epipolar plane with the image planes
- Corresponding points are on “conjugate” epipolar lines

Epipolar Constraint:



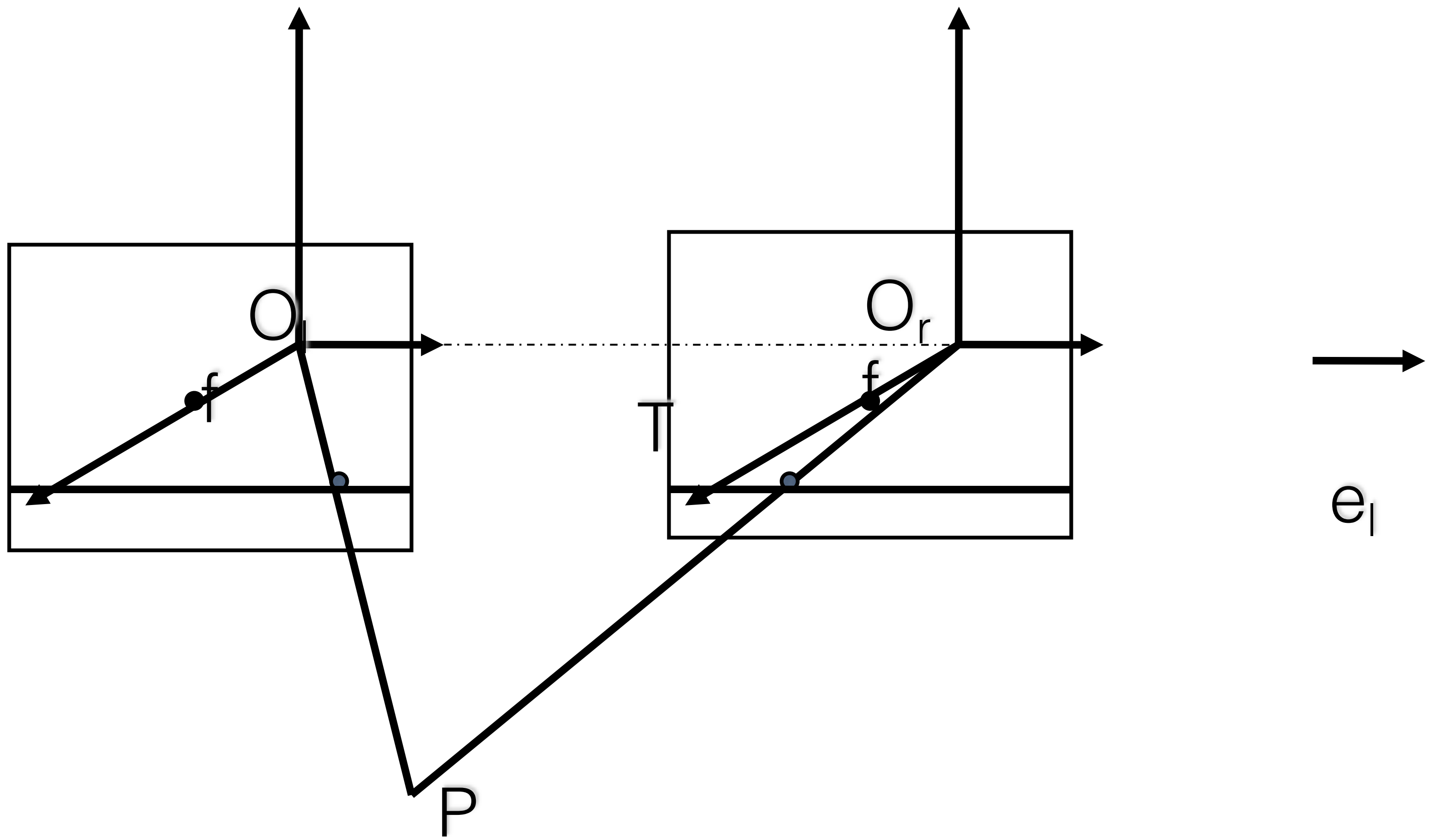
Find Epipoles:

- e_l: left image of O_r
- e_r: right image of O_l

Given p_l:

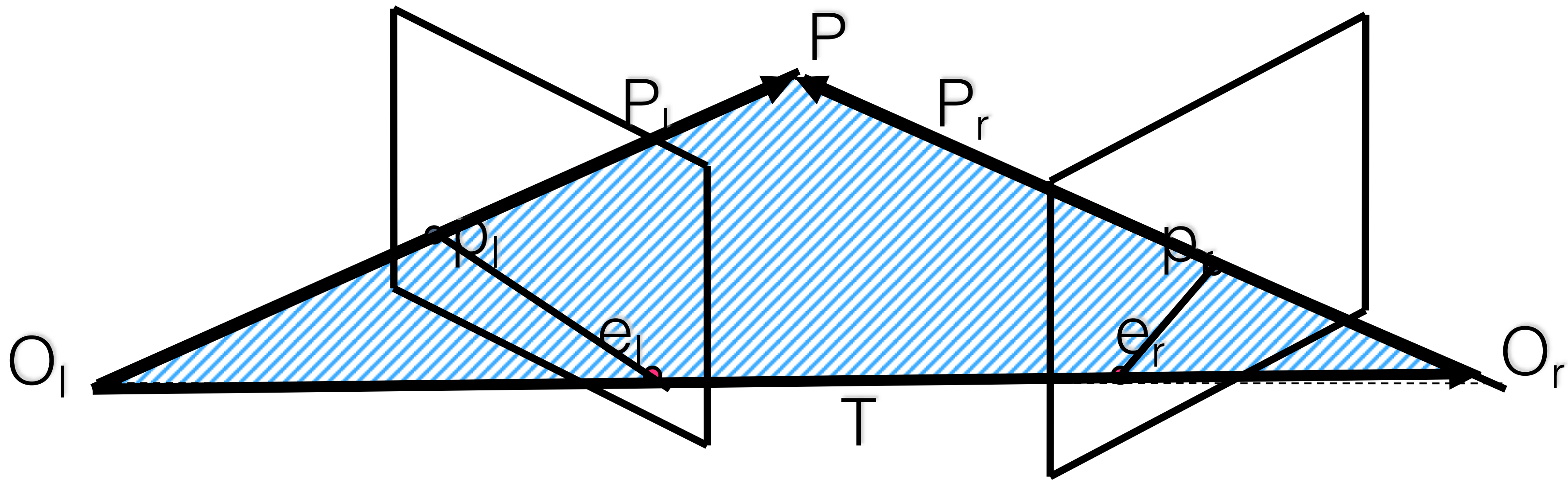
- consider its epipolar line: p_l e_l
- find epipolar plane: O_l,p_l,e_l
- intersect the epipolar plane with the right image plane
- search for p_r on the right epipolar line

Epipolar Geometry for Parallel Cameras



Epipoles are at infinity
Epipolar lines are parallel to the baseline

Essential Matrix

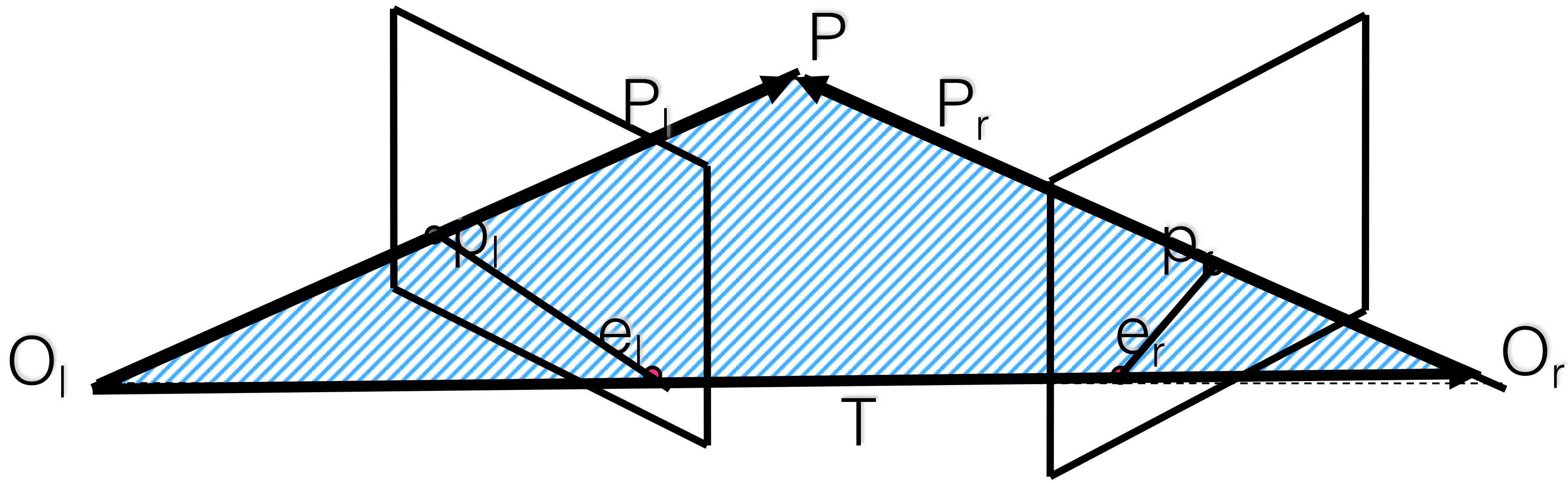


$$P_r = R(P_l - T)$$

$$P_l - T = R^{-1} P_r = R^T P_r$$

Essential Matrix

Epipolar constraint: P_l, T and P_l - T are coplanar:

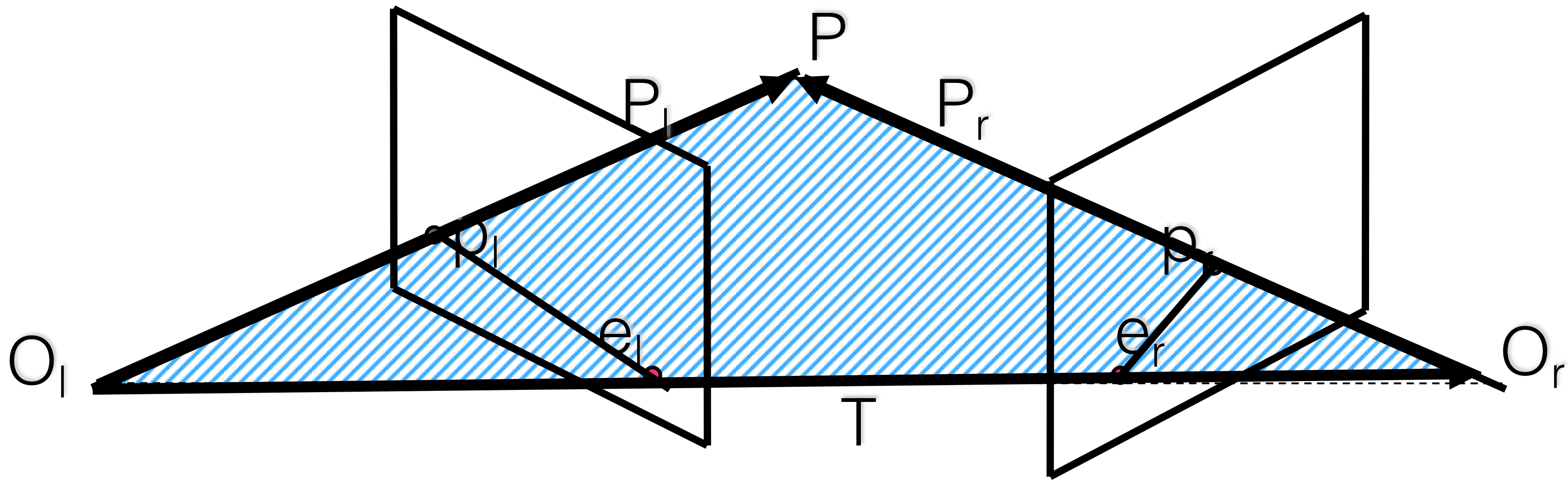


$$(P_l - T)^T \cdot T \times P_l = 0$$

$$P_l - T = R^T P_r \quad \Rightarrow (R^T P_r)^T \cdot T \times P_l = 0$$

Essential Matrix

Epipolar constraint: P_l, T and P_l - T are coplanar:



$$(R^T P_r)^T \cdot T \times P_l = 0$$

$$(P_r^T R) \cdot (T \times P_l) = 0$$

Vector Product as a Matrix Multiplication

$$T \times P_l = \begin{vmatrix} i & j & k \\ T_x & T_y & T_z \\ P_{l_x} & P_{l_y} & P_{l_z} \end{vmatrix}$$

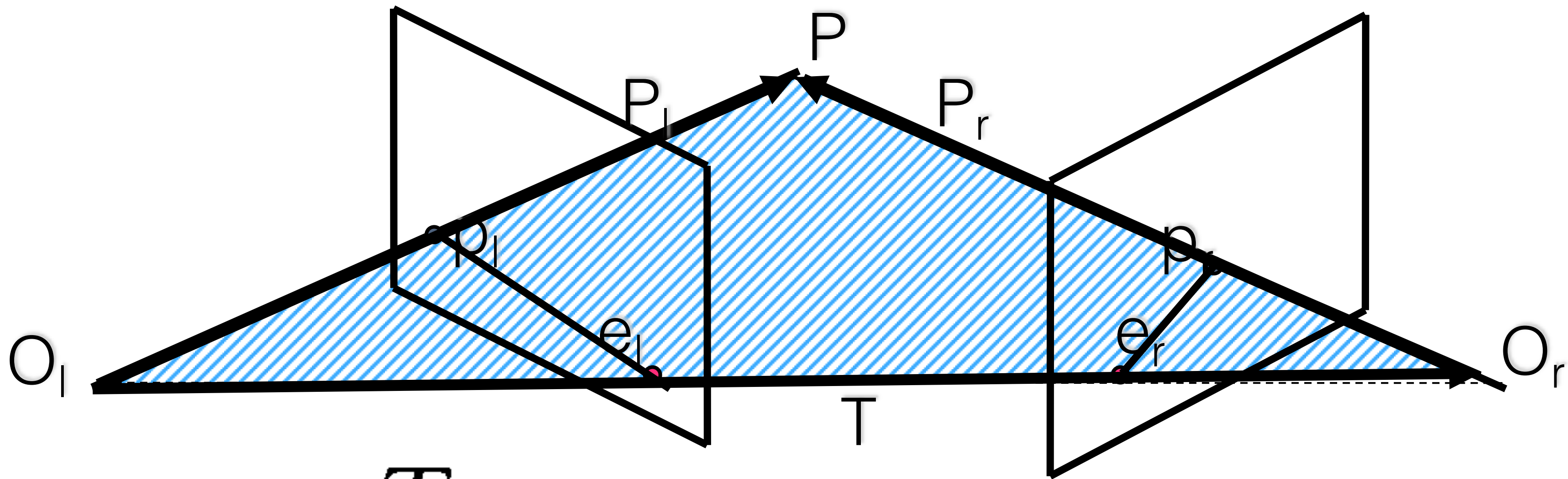
$$T \times P_l = (T_y P_{l_z} - T_z P_{l_y})i + (T_z P_{l_x} - T_x P_{l_z})j + (T_x P_{l_y} - T_y P_{l_x})k$$

$$T \times P_l = S P_l = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \begin{bmatrix} P_{l_x} \\ P_{l_y} \\ P_{l_z} \end{bmatrix} = \begin{bmatrix} T_y P_{l_z} - T_z P_{l_y} \\ T_z P_{l_x} - T_x P_{l_z} \\ T_x P_{l_y} - T_y P_{l_x} \end{bmatrix}$$

S has rank 2 ; it depends only on T

Essential Matrix

Epipolar constraint: P_l, T and P_l - T are coplanar:

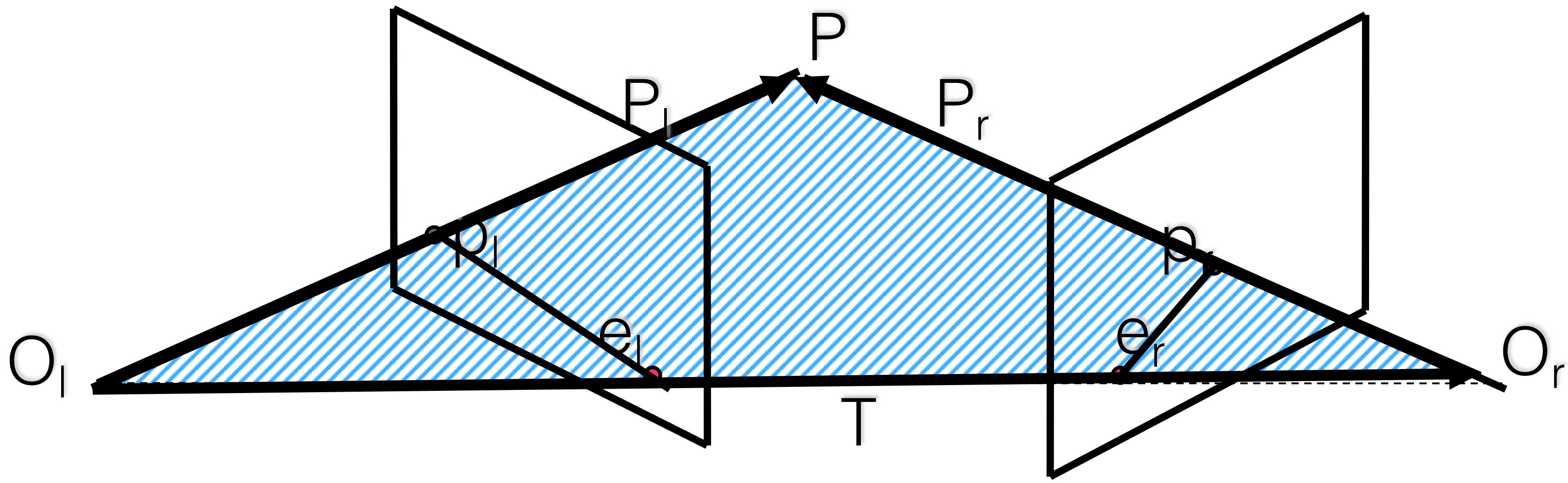


$$(P_r^T R) \cdot (T \times P_l) = 0$$

$$P_r^T R S P_l = 0$$

Essential Matrix

Epipolar constraint: P_l, T and P_l - T are coplanar:



$$P_r^T R S P_l = 0$$

Essential Matrix:

$$E = RS \qquad P_r^T E P_l = 0$$

Essential Matrix Properties

has rank 2

depends only on the EXTRINSIC Parameters (R & T)

$$E = RS$$

Longuet-Higgins equation

$$P_r^T E P_l = 0$$

$$p_l = \frac{f_l}{Z_l} P_l \qquad p_r = \frac{f_r}{Z_r} P_r$$

$$(\cancel{\frac{Z_r}{f_r}} p_r)^T E (\cancel{\frac{Z_l}{f_l}} p_l) = 0$$

$$\boxed{p_r^T E p_l = 0}$$

Epipolar Lines

Let l be a line in the image:

$$au + bv + c = 0$$

- Using homogeneous coordinates:

$$\tilde{p} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad \tilde{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \boxed{\tilde{p}^T \tilde{l} = \tilde{l}^T \tilde{p} = 0}$$

Epipolar Lines

Remember:

$$p_r^T E p_l = 0$$

$$\tilde{l}_r = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

p_r belongs to epipolar line in the right image defined by

$$\tilde{l}_r = E p_l$$

Epipolar Lines

Remember:

$$p_r^T E p_l = 0$$

$$\tilde{l}_l{}^T = \begin{bmatrix} a \\ b \\ c \end{bmatrix}^T$$

p_l belongs to epipolar line in the left image defined by

$$\tilde{l}_l = E^T p_r$$

Epipoles

Remember: epipoles belong to the epipolar lines

$$e_r^T E p_l = 0 \qquad\qquad p_r^T E e_l = 0$$

- And they belong to all the epipolar lines

$$e_r^T E = 0 \qquad\qquad E e_l = 0$$

Essential Matrix Summary

Longuet-Higgins equation $$p_r^T E p_l = 0$$

Epipolar lines:

$$\tilde{p_r}^T \tilde{l_r} = 0 \qquad \tilde{p_l}^T \tilde{l_l} = 0$$

$$\tilde{l_r} = E p_l \qquad \tilde{l_l} = E^T p_r$$

Epipoles:

$$e_r^T E = 0 \qquad E e_l = 0$$

Fundamental Matrix

The essential matrix uses **CAMERA** coordinates

To use image coordinates we must consider the INTRINSIC camera parameters:

$$\bar{p}_l = M_l p_l \qquad p_l = M_l^{-1} \bar{p}_l$$

$$\bar{p}_r = M_r p_r \qquad p_r = M_r^{-1} \bar{p}_r$$

Fundamental Matrix

$$p_l = M_l^{-1}\bar{p}_l$$

$$p_r = M_r^{-1}\bar{p}_r$$

$$p_r^T E p_l = 0$$

$$(M_r^{-1}\bar{p}_r)^T E (M_l^{-1}\bar{p}_l) = 0$$

$$\bar{p}_r^T (M_r^{-T} E M_l^{-1}) \bar{p}_l = 0$$

$$\boxed{\bar{p}_r^T F \bar{p}_l = 0}$$

Fundamental Matrix Properties

has rank 2

depends on the INTRINSIC and EXTRINSIC Parameters (f, etc ; R & T)

$$F = M_r^{-T} R S M_l^{-1}$$

Analogous to the Essential matrix, the Fundamental matrix also tells how points in each image are related to epipolar lines in the other image.

Computing F: The 8 pt Algorithm

Assume that you have m correspondences

Each correspondence satisfies:

$$\bar{p}_{r_i}^T F \bar{p}_{l_i} = 0 \quad i = 1, \ldots, m$$

- F is a 3x3 matrix (9 entries) but rank 2
- **HOMOGENEOUS** linear system with 9 unknowns
- Need m >= 8; solution will be up to a constant

Computing F: The 8 pt Algorithm

$$\bar{p}_{l_i} = (x_i \ y_i \ 1)^T \quad \bar{p}_{r_i} = (x_i' \ y_i' \ 1)^T$$

$$\bar{p}_{r_i}^T F \bar{p}_{l_i} = 0 \quad i = 1, \ldots, m$$

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

Computing F: The 8 pt Algorithm

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$x_i x'_i f_{11} + x_i y'_i f_{21} + x_i f_{31} +$$
$$y_i x'_i f_{12} + y_i y'_i f_{22} + y_i f_{32} +$$
$$x'_i f_{13} + y'_i f_{23} + f_{33} = 0$$

Computing F: The 8 pt Algorithm

$$\begin{bmatrix} x_i' & y_i' & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} x_1 x_1' & x_1 y_1' & x_1 & y_1 x_1' & y_1 y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_m x_m' & x_m y_m' & x_m & y_m x_m' & y_m y_m' & y_m & x_m' & y_m' & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

Solving Homogeneous Systems

Assume that we need to find the non trivial solution of:

$$A\mathbf{x} = 0$$

with m equations and n unknowns, m >= n – 1 and rank(A) = n-1

Since the norm of x is arbitrary, we will look for a solution with norm ||x|| = 1

Least Square solution

We want Ax as close to 0 as possible and ||x|| =1:

$$\min_{\mathbf{x}} ||A\mathbf{x}||^2 \text{ s.t. } ||x||^2 = 1$$

$$||A\mathbf{x}||^2 = (A\mathbf{x})^T(A\mathbf{x}) = \mathbf{x}^T A^T A\mathbf{x}$$

$$||\mathbf{x}||^2 = \mathbf{x}^T\mathbf{x} = 1$$

Optimization with constraints

Define the following cost:

$$\mathcal{L}(\mathbf{x}) = \mathbf{x}^T A^T A \mathbf{x} - \lambda(\mathbf{x}^T \mathbf{x} - 1)$$

This cost is called the LAGRANGIAN cost and λ is called the LAGRANGIAN multiplier

The Lagrangian incorporates the constraints into the cost function by introducing extra variables.

Optimization with constraints

$$\min_{\mathbf{x}} \left\{ \mathcal{L}(\mathbf{x}) = \mathbf{x}^T A^T A\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - 1) \right\}$$

Taking derivatives wrt to x and λ:

$$A^T A\mathbf{x} - \lambda\mathbf{x} = 0$$
$$\mathbf{x}^T\mathbf{x} - 1 = 0$$

- The first equation is an eigenvector problem
- The second equation is the original constraint

Optimization with constraints

$$A^T A\mathbf{x} - \lambda\mathbf{x} = 0$$
$$A^T A\mathbf{x} = \lambda\mathbf{x}$$

- x is an eigenvector of A^TA with eigenvalue λ: e_λ

$$\mathcal{L}(\mathbf{e}_\lambda) = {\mathbf{e}_\lambda}^T A^T A\mathbf{e}_\lambda - \lambda({\mathbf{e}_\lambda}^T\mathbf{e}_\lambda - 1)$$
$$\mathcal{L}(\mathbf{e}_\lambda) = \lambda{\mathbf{e}_\lambda}^T\mathbf{e}_\lambda = \lambda$$

- We want the eigenvector with <u>smallest</u> eigenvalue

We can find the eigenvectors and eigenvalues of A^TA by finding the Singular Value Decomposition of A

Singular Value Decomposition (SVD)

Any m x n matrix A can be written as the product of 3 matrices:

$$A = UDV^T$$

Where:

- U is m x m and its columns are orthonormal vectors
- V is n x n and its columns are orthonormal vectors
- D is m x n diagonal and its diagonal elements are called the singular values of A, and are such that:

$$\sigma_1 >= \sigma_2 >= \dots \sigma_n >= 0$$

SVD Properties

$$A = UDV^T$$

- The columns of U are the eigenvectors of AA^T
- The columns of V (rows of V^T) are the eigenvectors of A^TA
- The squares of the diagonal elements of D are the eigenvalues of AA^T and A^TA

Computing F: The 8 pt Algorithm

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

$$A\mathbf{x} = 0$$

A has rank 8

$$\min_{\mathbf{x}} ||A\mathbf{x}||^2 \text{ s.t. } ||x||^2 = 1$$

- Find the eigenvector of A^TA with smallest eigenvalue!

Algorithm EIGHT_POINT

The input is formed by m point correspondences, m >= 8

Construct the m x 9 matrix A

Find the SVD of A: $A = UDV^T$

The columns of V are the eigenvectors of A^TA; the last one corresponds to the smallest eigenvalue:

  The entries of F are the components of the last column of V corresponding to the least s.v.

Algorithm EIGHT_POINT

F must be singular. To enforce it:

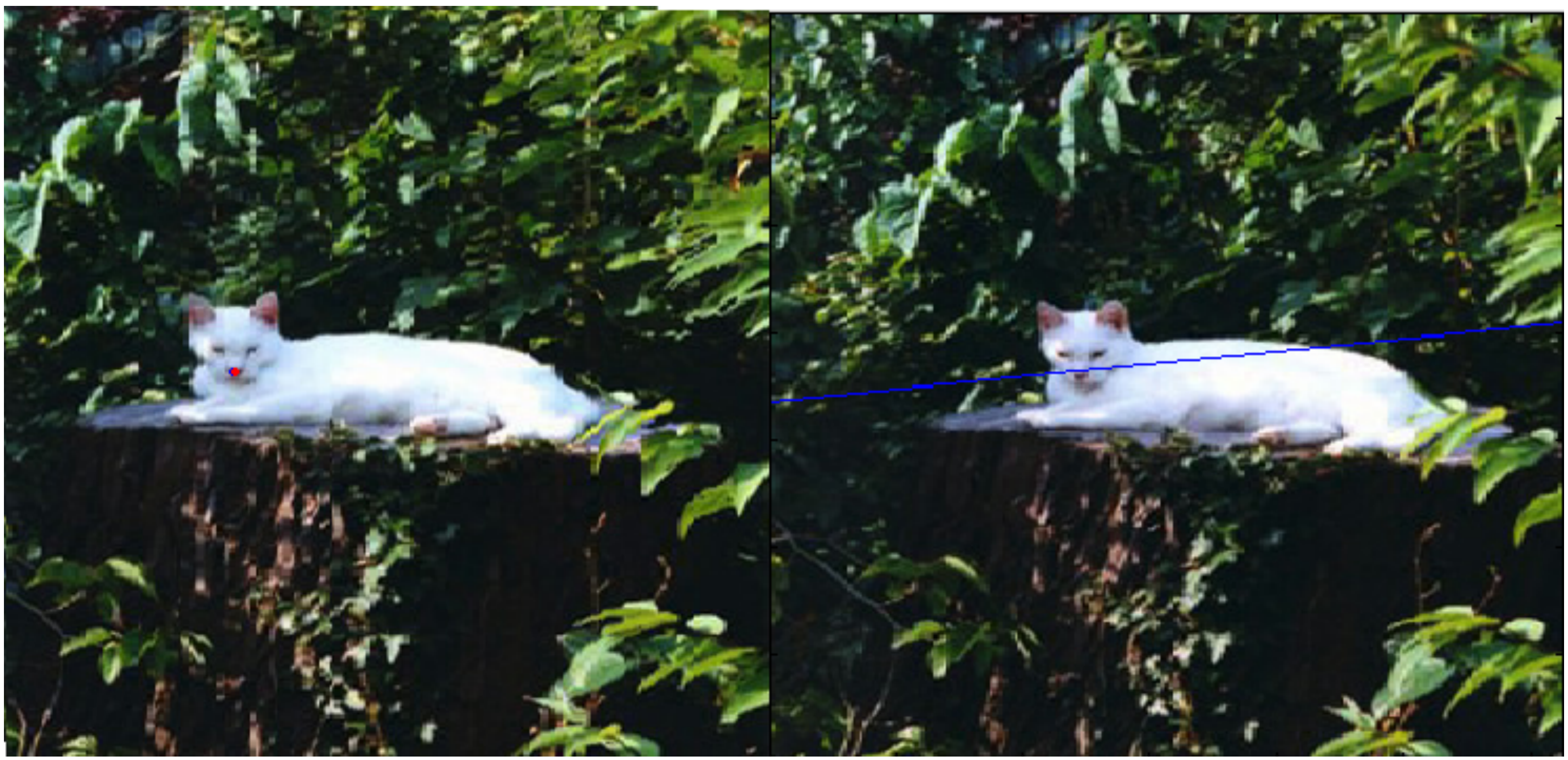
Find the SVD of F: $F = U_f D_f V_f^T$

Set smallest s.v. of F to 0 to create D'_f

Recompute F: $F = U_f D'_f V_f^T$

Example

Example



Numerical Details

The coordinates of corresponding points can have a wide range leading to numerical instabilities.
It is better to first normalize them so they have average 0 and stdev 1 and de-normalize F at the end:

$$\hat{x}_i = T x_i \qquad \hat{x}'_i = T' x'_i \qquad T = \begin{bmatrix} \frac{1}{\sigma^2} & 0 & -\mu_x \\ 0 & \frac{1}{\sigma^2} & -\mu_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{F = T'^{T} F_n T}$$

Image Rectification

Assuming extrinsic parameters R & T are known, compute the image transformation that makes conjugate epipolar lines collinear and parallel to the horizontal image axis

Image Rectification

Rectification involves two rotations:

- First rotation sends epipole to infinity along horizontal axis
- Second rotation makes epipolar lines parallel

Rotate the left and right cameras with first R_1

Rotate the right camera with the R matrix

Adjust scales in both camera references

Image Rectification

Build the first rotation:

$$R_{rect} = \begin{bmatrix} \mathbf{e_1}^T \\ \mathbf{e_2}^T \\ \mathbf{e_3}^T \end{bmatrix}$$

with: $\mathbf{e_1} = \dfrac{\mathbf{T}}{||\mathbf{T}||}$ makes epipole go to the hor. axis

$$\mathbf{e_2} = \frac{1}{\sqrt{T_x^2 + T_y^2}} \begin{bmatrix} -T_y \\ T_x \\ 0 \end{bmatrix}$$

orthogonal to optical axis

$$\mathbf{e_3} = \mathbf{e_1} \times \mathbf{e_2}$$

Algorithm Rectification

Build the matrix R_{rect}

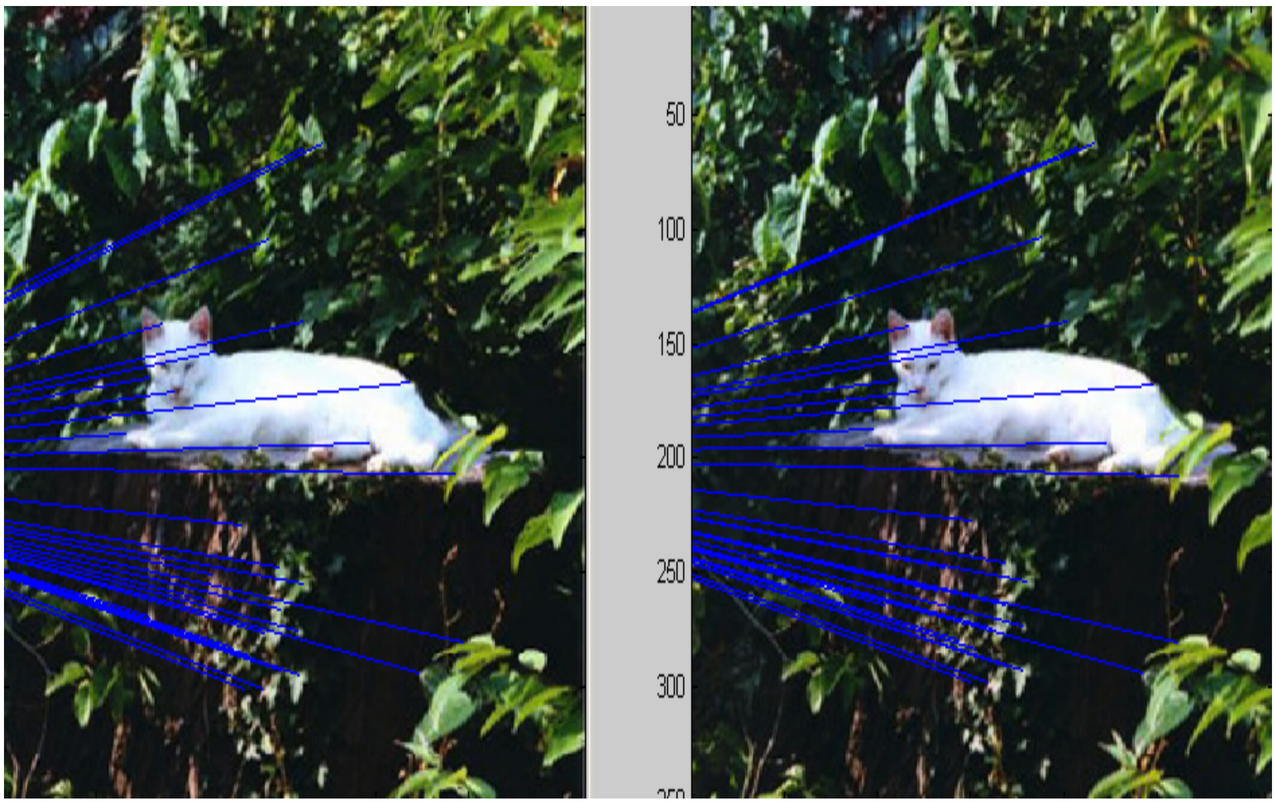
Set $R_l = R_{rect}$ and $R_r = R.R_{rect}$

For each left point $p_l = (x,y,f)^T$

- compute $R_l\ p_l = (x', y', z')^T$
- Compute $p'_l = f/z'\ (x', y'\ z')^T$

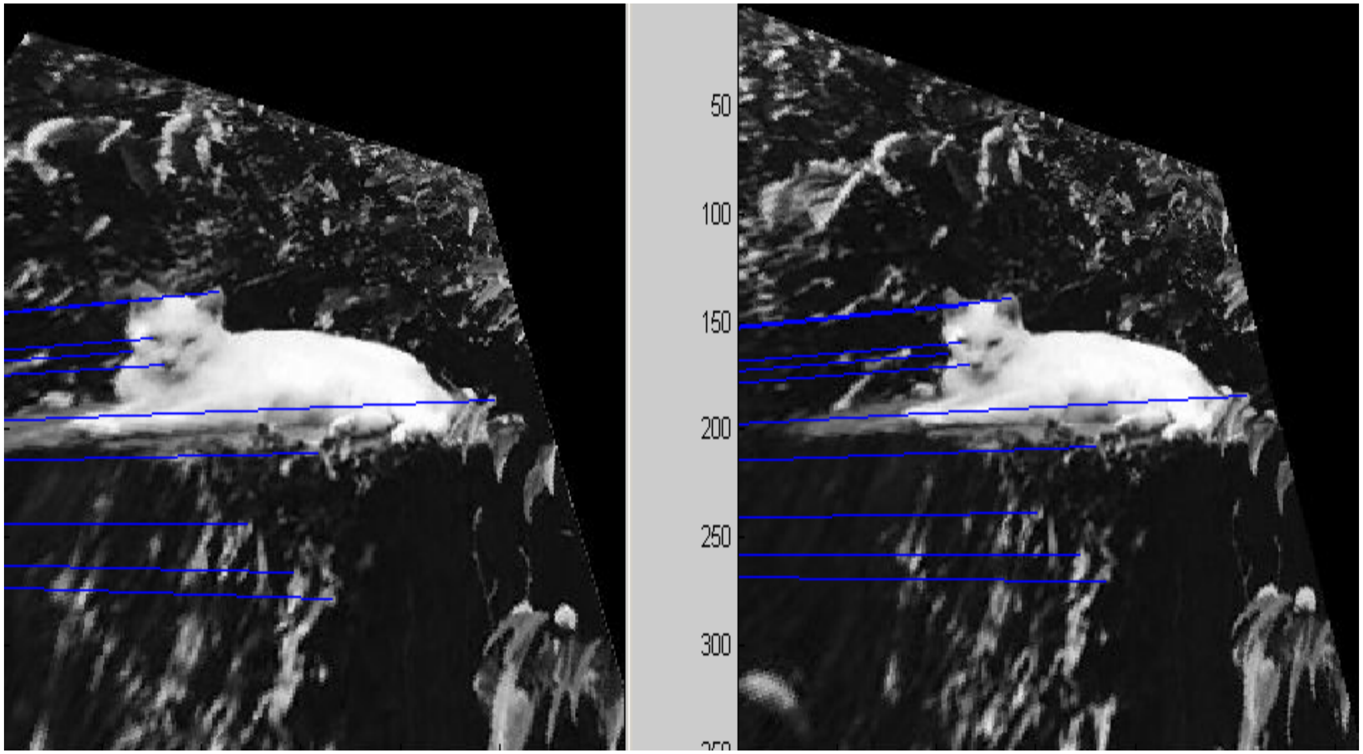
Repeat above for the right camera with R_r and p_r

Image Rectification



Stereo Images prior to rectification

Image rectification



Stereo Images after rectification

Image Rectification

3D Reconstruction

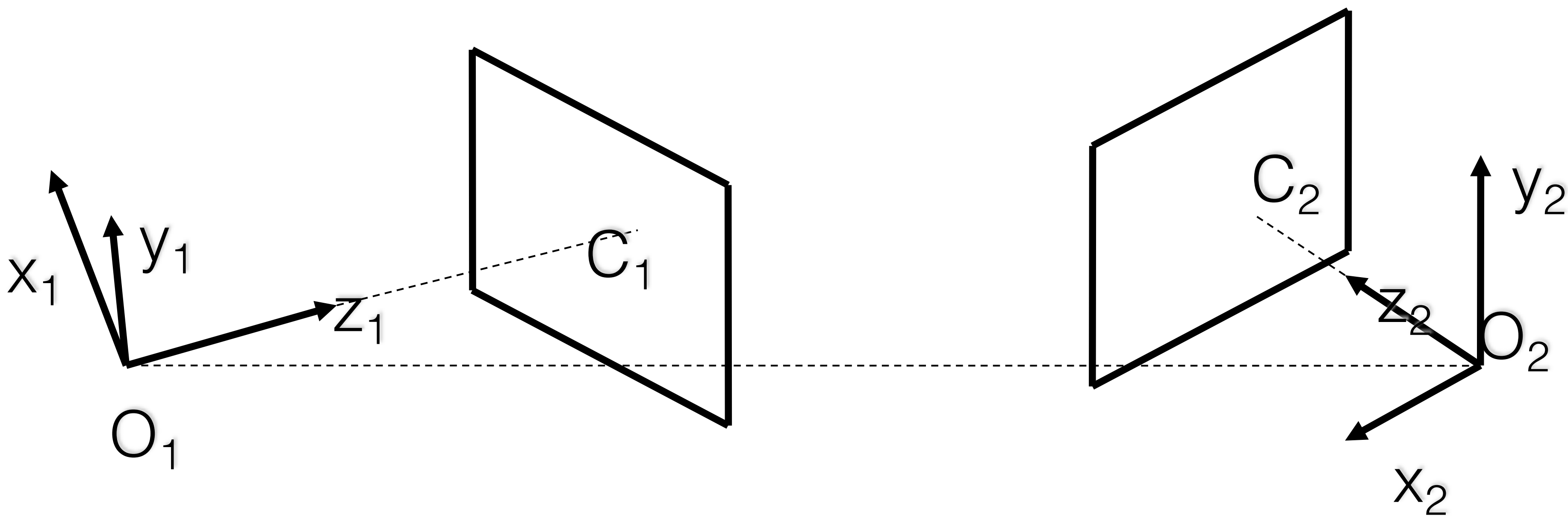
3D Reconstruction

3 cases:

- Fully calibrated: intrinsic & extrinsic parameters are known.
  - UNAMBIGUOUS reconstruction
- Only intrinsic parameters are known.
  - Up to a SCALE FACTOR
- No parameters are known.
  - Up to an UNKNOWN PROJECTIVE transformation

FULLY CALIBRATED

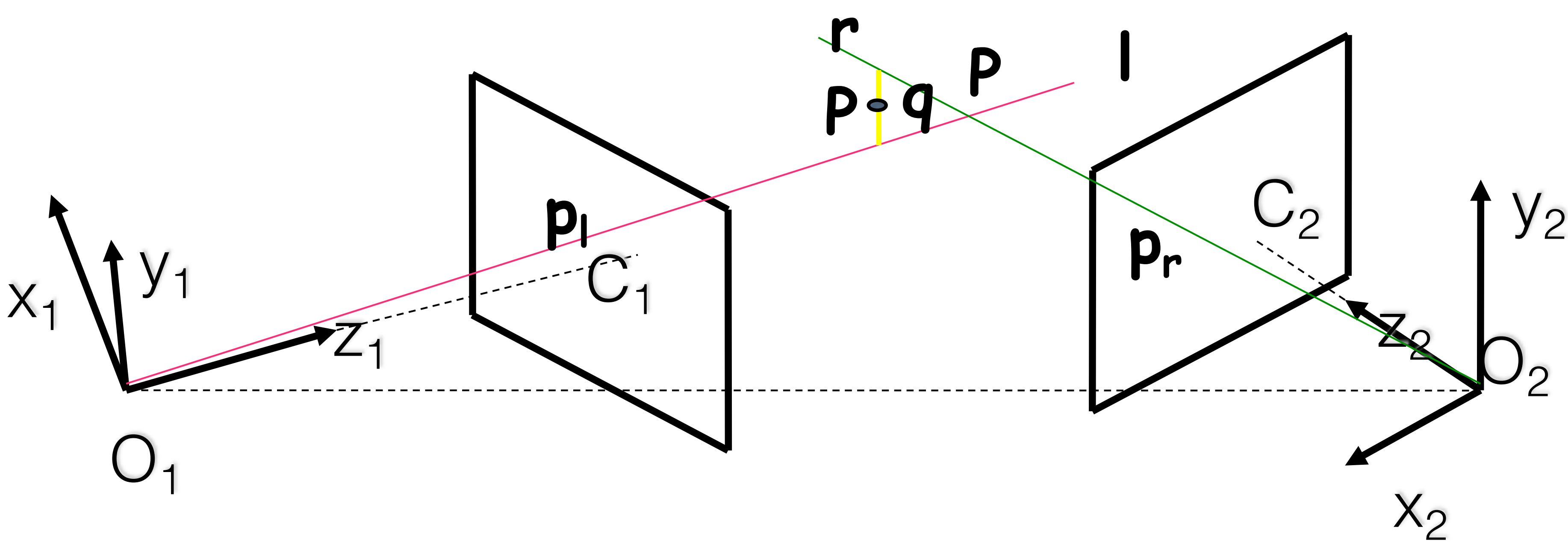
Parameters of a Stereo System



- Intrinsic:
- f_1 and f_2: focal lengths
- c_1 and c_2: principal points
- Pixel size

- Extrinsic
- Transformation (R,T) between cameras

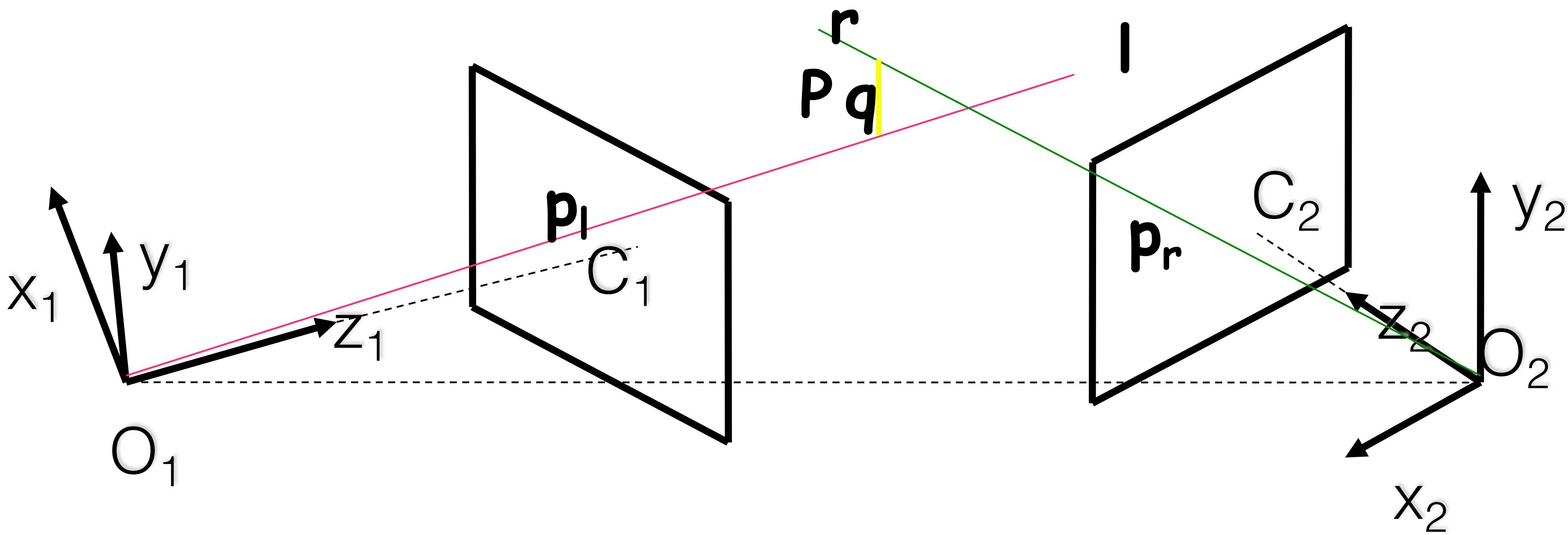
3D Reconstruction



- P should lie in the intersection of the 2 rays r and l

- Due to numerical errors, correspondences errors, noise, etc, the rays may not intersect!

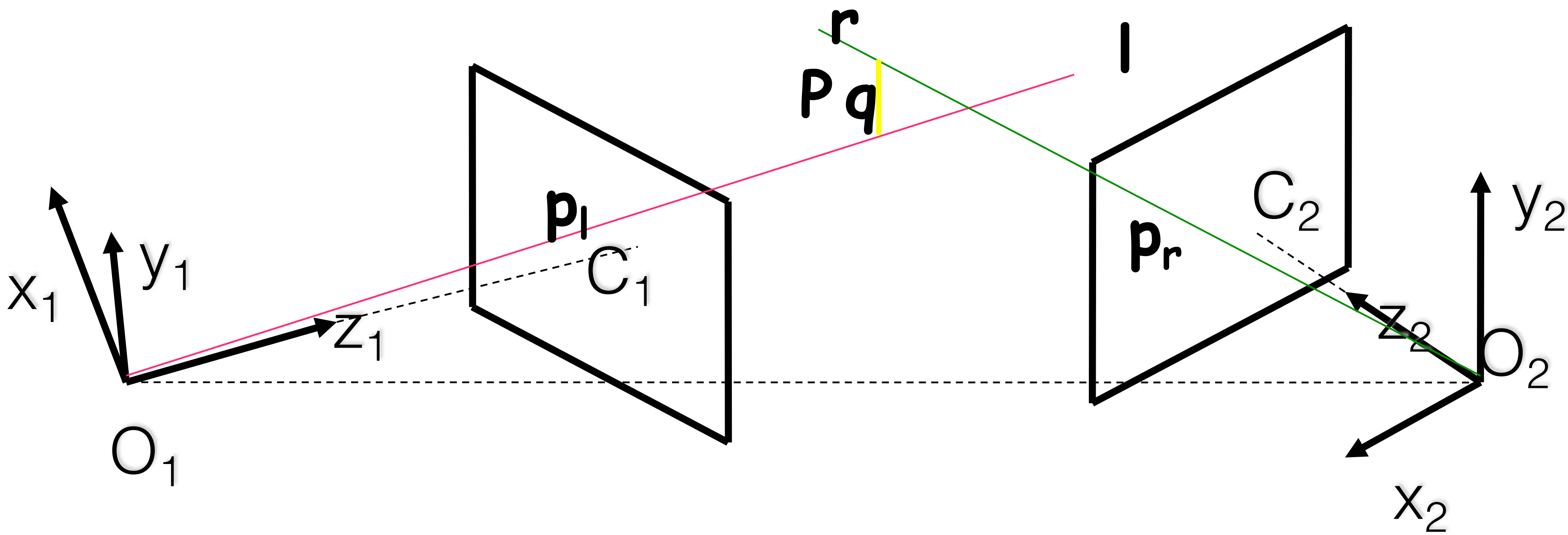
3D Reconstruction



$$T = O_r - O_l$$

$$P_r = R(P_l - T)$$

3D Reconstruction



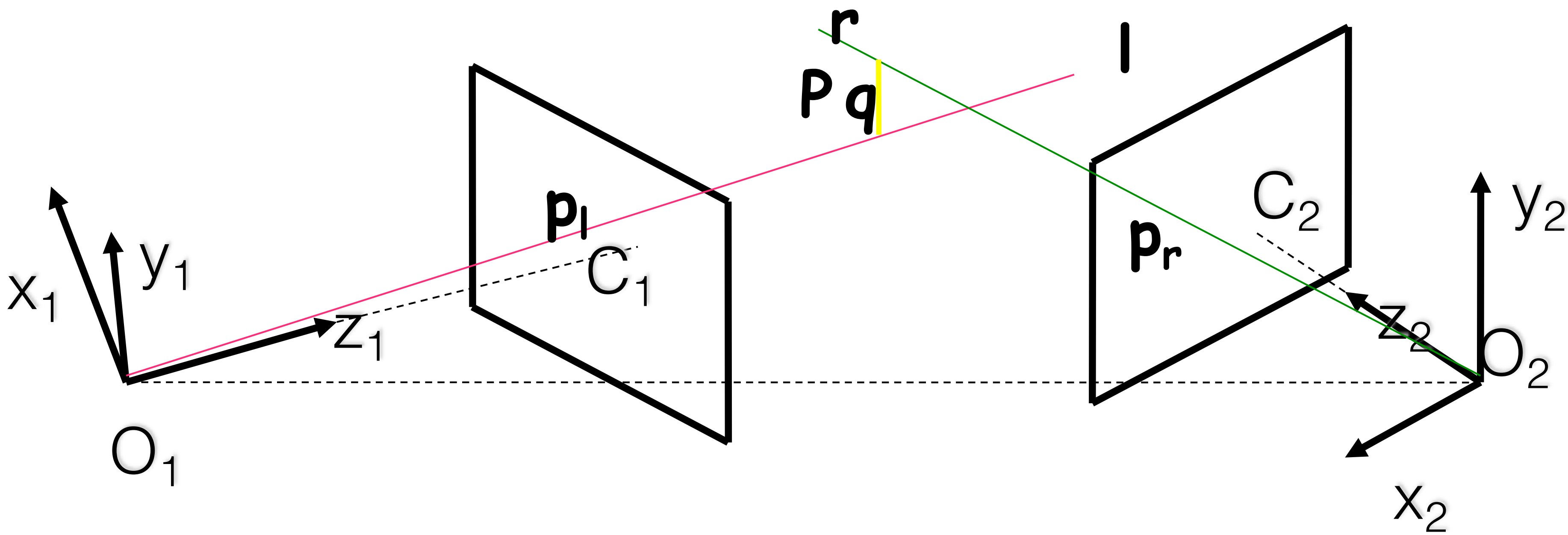
$$
\begin{aligned}
T &= O_r - O_l \\
P_r &= R(P_l - T)
\end{aligned}
$$

Ray l:

$$p = a.p_l$$

a is a real number

3D Reconstruction



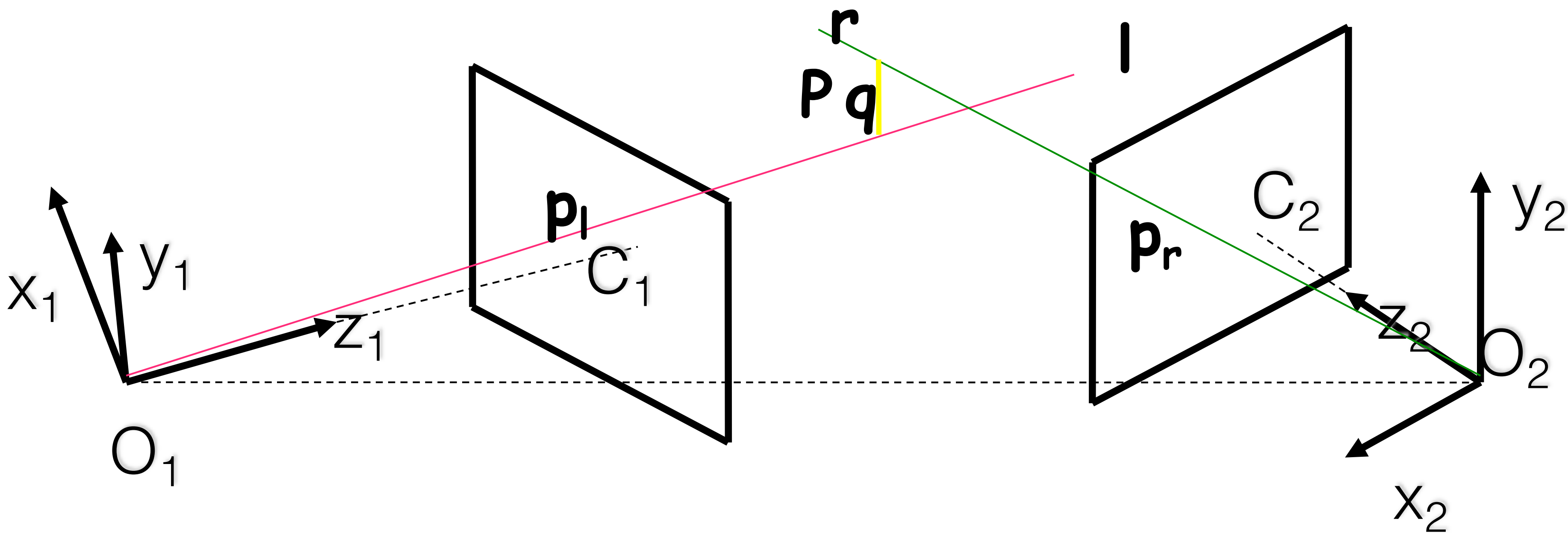
$$
\begin{aligned}
T &= O_r - O_l \\
P_r &= R(P_l - T)
\end{aligned}
$$

Ray r:

$$p = T + b.R^T p_r$$

b is a real number

3D Reconstruction



Ray l:

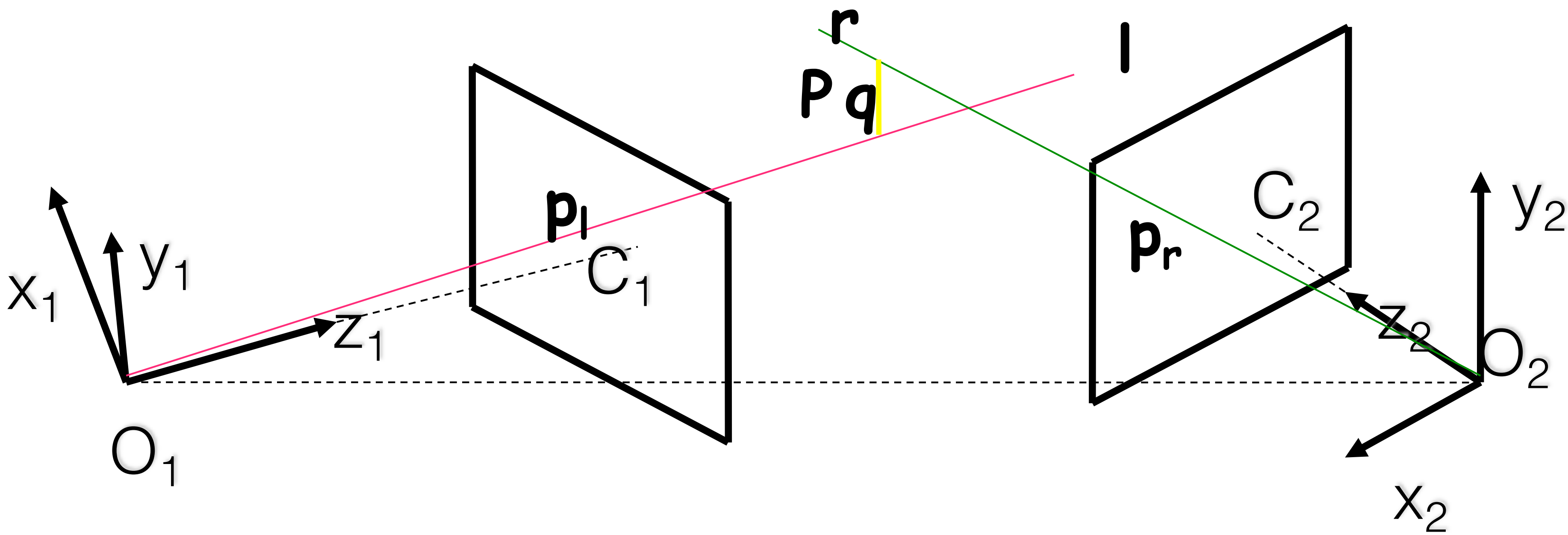
$$p = a.p_l$$

Ray r:

$$p = T + b.R^T p_r$$

Consider now a line PERPENDICULAR to both

3D Reconstruction



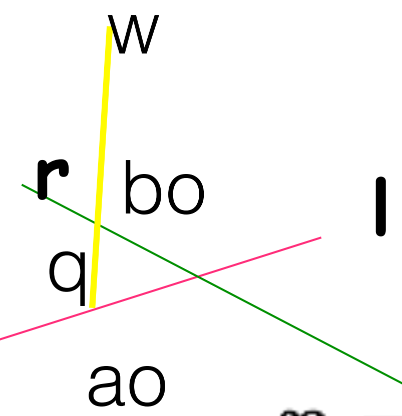
Ray l:

Ray r:

$$p = a.p_l \qquad\qquad p = T + b.R^T p_r$$

$$w \sim c.(p_l \times R^T p_r) \quad w \perp r \;\; ; w \perp l$$

3D Reconstruction



Ray l:

$$p = a.p_l$$

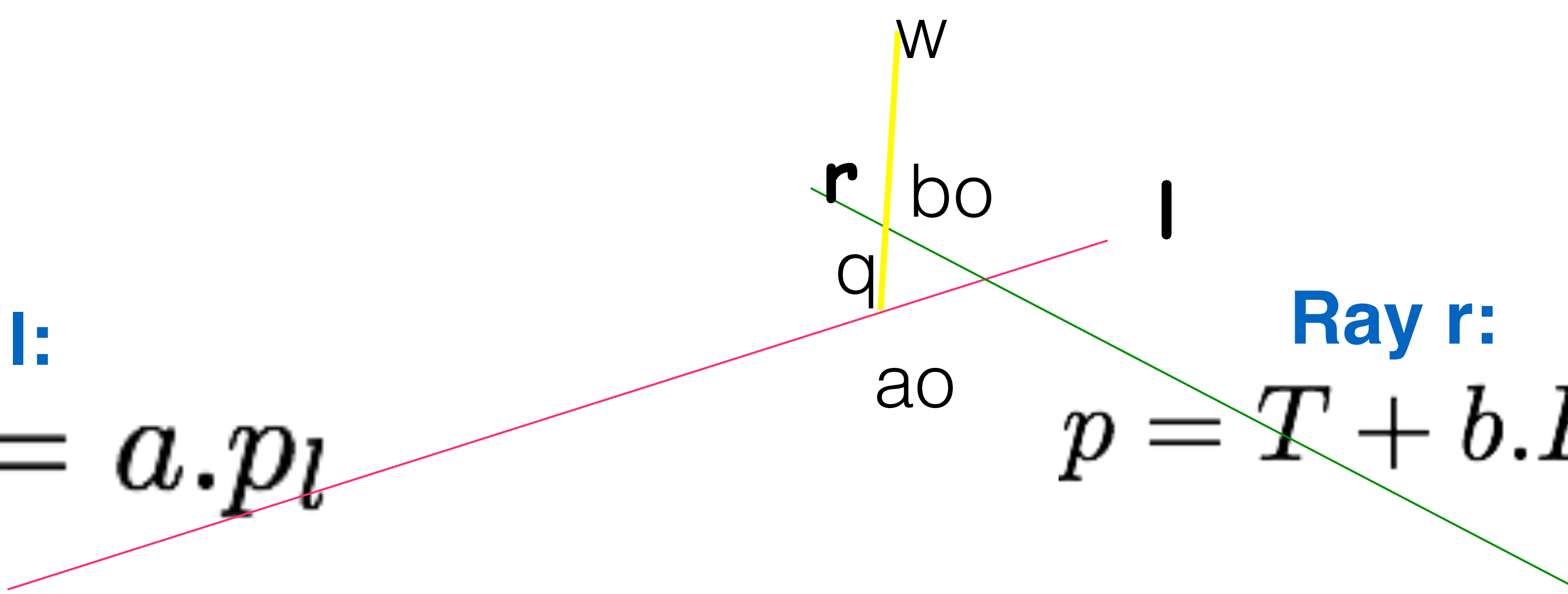
Ray r:

$$p = T + b.R^T p_r$$

$$w \sim c.(p_l \times R^T p_r) \quad w \perp r \;\; ; w \perp l$$

$$a_o p_l + \underbrace{c_o(p_l \times R^T p_r)}_{q} = T + b_o R^T p_r$$

3D Reconstruction



Ray l:

$$p = a.p_l$$

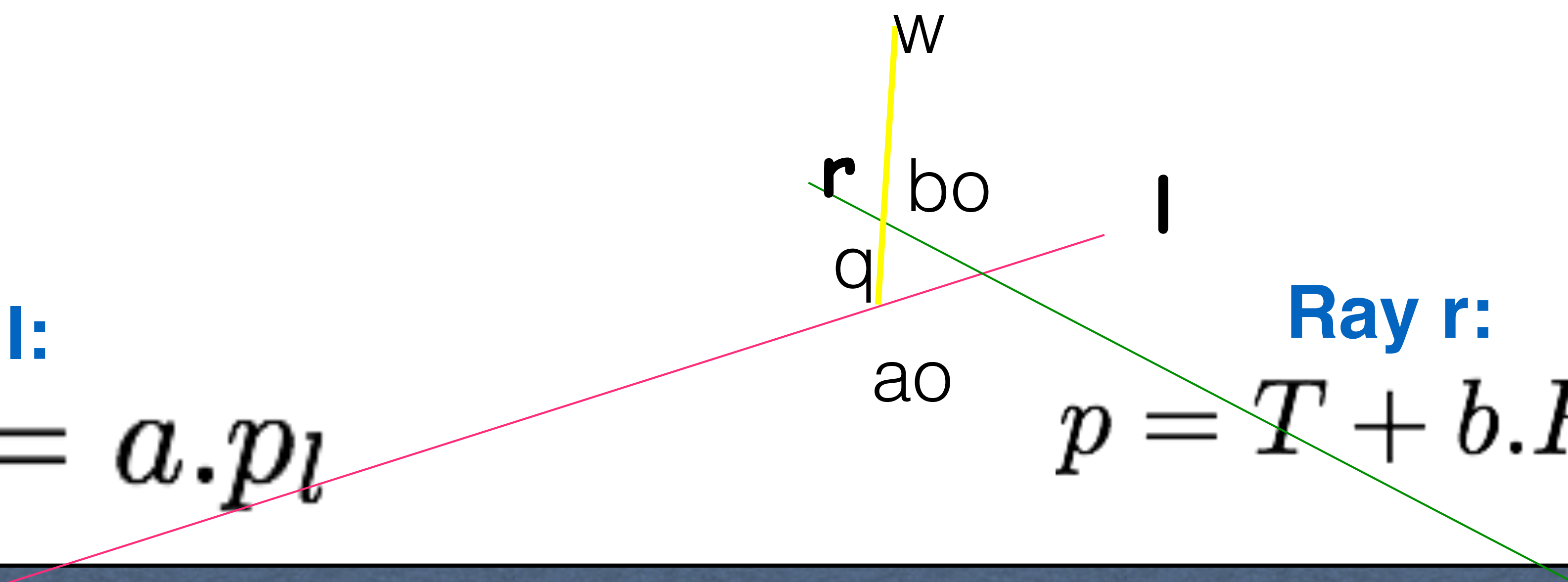
Ray r:

$$p = T + b.R^T p_r$$

$$a_o p_l + c_o (p_l \times R^T p_r) = T + b_o R^T p_r$$

$$\boxed{a_o p_l - b_o R^T p_r + c_o (p_l \times R^T p_r) = T}$$

3D Reconstruction



Ray l:

$$p = a.p_l$$

Ray r:

$$p = T + b.R^T p_r$$

$$a_o p_l - b_o R^T p_r + c_o (p_l \times R^T p_r) = T$$

Solve for a_o, b_o, and c_o. How?

Choose P as the Midpoint of segment $a_o b_o$

Only <u>**INTRINSIC**</u> parameters known

3D Reconstruction

Up to a Scale Factor

3D Reconstruction

Assume:

- Intrinsic parameters are known for both cameras
- N >= 8 correspondences are given;
- The Essential matrix was found from the above

Since we don’t know the BASELINE, we cannot recover the SCALE of the scene.

Reconstruction is up to a scale factor

3D Reconstruction

4 steps:

1) Recover a normalized translation vector T
   True orientation (up to a sign), but unit length
2) Recover rotation R
3) Reconstruct Zl and Zr for each point
4) Test the signs for consistency and to recover true sign of T

Step 1: Recover T (up to a scale)

Remember the Essential Matrix:

$$E = RS$$

R is the rotation matrix

$$T \times P_l = SP_l = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix} \begin{bmatrix} P_{l_x} \\ P_{l_y} \\ P_{l_z} \end{bmatrix} = \begin{bmatrix} T_y P_{l_z} - T_z P_{l_y} \\ T_z P_{l_x} - T_x P_{l_z} \\ T_x P_{l_y} - T_y P_{l_x} \end{bmatrix}$$

S has rank 2 ; it depends only on T

Step 1: Recover T (up to a scale)

$$E^T E = S^T \underbrace{R^T R}_{\textbf{Identity!}} S = S^T S$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}$$

$$S^T = \begin{bmatrix} 0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0 \end{bmatrix} \quad S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Compute the TRACE:

$$\mathrm{Trace}(S^T S) = 2T_x^2 + 2T_y^2 + 2T_z^2 = 2||T||^2$$

$$N = \sqrt{\mathrm{Trace}(S^T S)/2}$$

Normalization factor

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Normalize T by N: $N = \sqrt{\mathrm{Trace}(S^T S)/2}$

$$\hat{T} = \frac{T}{||T||} = \frac{T}{N}$$

Step 1: Recover T (up to a scale)

$$E^T E = S^T S$$

$$S^T S = \begin{bmatrix} T_y^2 + T_z^2 & -T_x T_y & -T_x T_z \\ -T_x T_y & T_z^2 + T_x^2 & -T_y T_z \\ -T_x T_z & -T_y T_z & T_x^2 + T_y^2 \end{bmatrix}$$

Using the normalized translation (divide by N the matrix above):

$$\hat{E}^T \hat{E} = \begin{bmatrix} 1 - \hat{T}_x^{\,2} & -\hat{T}_x \hat{T}_y & -\hat{T}_x T_z \\ -\hat{T}_y \hat{T}_x & 1 - \hat{T}_y^{\,2} & -\hat{T}_y \hat{T}_z \\ -\hat{T}_z \hat{T}_x & -\hat{T}_z \hat{T}_y & 1 - \hat{T}_z^{\,2} \end{bmatrix}$$

Step 1: Recover T (up to a scale)

$$\hat{E^T}\hat{E} = \begin{bmatrix} 1 - \hat{T_x}^2 & -\hat{T_x}\hat{T_y} & -\hat{T_x}T_z \\ -\hat{T_y}\hat{T_x} & 1 - \hat{T_y}^2 & -\hat{T_y}\hat{T_z} \\ -\hat{T_z}\hat{T_x} & -\hat{T_z}\hat{T_y} & 1 - \hat{T_z}^2 \end{bmatrix}$$

Find a unit vector $\hat{T}$

This can be done from any row or column. HOWEVER, only up to a sign since each entry is quadratic

Step 2: Find R

$$\hat{E}_i \text{ is the } i^{th} \text{ row of } \hat{E}$$

Define: $$w_i = \hat{E}_i \times \hat{T}$$

Then (Exercise!):

$$\begin{aligned} R_1 &= w_1 + w_2 \times w_3 \\ R_2 &= w_2 + w_3 \times w_1 \\ R_3 &= w_3 + w_1 \times w_2 \end{aligned}$$

Are the rows of R

Step 3: Find Zl and Zr

Given E (up to a sign factor)

  we can find T up to a sign (and scale factor)

  Once we have T we can find R

Therefore, we have 4 possibilities for (T,R), depending on the sign of E and T!

Step 3: Find Zl and Zr

Consider P expressed in the left and right coordinates: Pl and Pr

$$P_r = R(P_l - T)$$

And consider its left and right images: pl and pr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T (P_l - T)} \qquad p_l = f_l \frac{P_l}{Z_l}$$

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)} \qquad p_l = f_l \frac{P_l}{Z_l}$$

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

2

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)}$$

3

$$\boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)}$$

$$p_l = f_l \frac{P_l}{Z_l} \xrightarrow{5} \text{Pl}$$

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

2

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)}$$

3

$$\boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

4

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)}$$

$$p_l = f_l \frac{P_l}{Z_l} \xrightarrow{5} \text{Pl}$$

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

4

2

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)} \xrightarrow{3} \boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

Step 3: Find Zl and Zr

$$p_r = f_r \frac{R(P_l - T)}{R_3^T(P_l - T)}$$

$$p_l = f_l \frac{P_l}{Z_l} \xrightarrow{5} \text{Pl}$$

6

$$P_r = R(P_l - T)$$

$$\downarrow 7$$

Pr

1

$$x_r = f_r \frac{R_1^T(P_l - T)}{R_3^T(P_l - T)}$$

2

$$x_r = f_r \frac{R_1^T(Z_l p_l - f_l T)}{R_3^T(Z_l p_l - f_l T)} \xrightarrow{3} \boxed{Z_l = f_l \frac{(f_r R_1 - x_r R_3)^T T}{(f_r R_1 - x_r R_3)^T p_l}}$$

4

Step 4: Check the sign of Zl and Zr

We had 4 possible pairs (R,T) and for each pair we can compute Zl and Zr

ONLY one set yields Zl and Zr positive for ALL the points:

- If BOTH < 0 for some pt
  - Change the sign of T, recompute depths
- If ONE < 0 for some pt
  - Change the sign of each entry in E, recompute R and depths
- If BOTH > 0 we have the right (R,T)

NO CALIBRATION

3D Reconstruction

Up to a PROJECTIVITY

Projective Ambiguity

Fig. 1.4. **Projective ambiguity:** *Reconstructions of a mug (shown with the true shape in the centre) under 3D projective transformations in the Z direction. Five examples of the cup with different degrees of projective distortion are shown. The shapes are quite different from the original.*

3D Reconstruction

Assume:

No calibration is known

N >= 8 correspondences are given;

The Fundamental matrix was found from the above

Reconstruction is up to a projectivity

3D Reconstruction

We can recover the projection matrix for each camera up to an unknown projectivity.

Once we have these matrices, we can triangulate in projective space.

2D Projectivity

Consider the projective space P^2

Pts have THREE coordinates (X,Y,1)'

Recall: Given 4 correspondences between points (not three colinear) we can find the projectivity (up to a constant) between the points.

STANDARD BASIS PTS in P2

$$\begin{aligned} P_1 &= [1, 0, 0]^T \\ P_2 &= [0, 1, 0]^T \\ P_3 &= [0, 0, 1]^T \\ P_4 &= [1, 1, 1]^T \end{aligned}$$

3D Projectivity

Consider the projective space P^3

Pts have FOUR coordinates (X,Y,Z,1)’

Recall: Given 5 correspondences between points (not three colinear, not four coplanar) we can find the projectivity (up to a constant) between the points.

STANDARD BASIS PTS in P3

$$
\begin{aligned}
P_1 &= [1,0,0,0]^T \\
P_2 &= [0,1,0,0]^T \\
P_3 &= [0,0,1,0]^T \\
P_4 &= [0,0,0,1]^T \\
P_5 &= [1,1,1,1]^T
\end{aligned}
$$

3D Reconstruction

Choose 4 pts in one image and choose them as a basis for p2 to p2

Corresponding pts in 3D plus one more are chosen as basis for p3 to p3

Makes math easier (lots of zeros!)

Solve for M: matrix projection between p in p2 and P in P3

Impose that epipoles are in the null space of M.

Let M be the projection matrix for the left image.

Consider first 5 pts where no 3 collinear, 4 no coplanar as the standard basis sent to the standard basis plus one more point:

$$\begin{aligned} P_1 &= [1,0,0,0]^T \\ P_2 &= [0,1,0,0]^T \\ P_3 &= [0,0,1,0]^T \\ P_4 &= [0,0,0,1]^T \\ P_5 &= [1,1,1,1]^T \end{aligned}$$

$$MP_i = \rho_i p_i$$

$$\begin{aligned} p_1 &= [1,0,0]^T \\ p_2 &= [0,1,0]^T \\ p_3 &= [0,0,1]^T \\ p_4 &= [1,1,1]^T \end{aligned}$$

$$MP_1 = \rho_1 p_1$$

$$\begin{bmatrix} m11 & m12 & m13 & m14 \\ m21 & m22 & m23 & m24 \\ m31 & m32 & m33 & m34 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \rho_1 \\ 0 \\ 0 \end{bmatrix}$$

$$MP_4 = \rho_4 p_4$$

$$\begin{bmatrix} m11 & m12 & m13 & m14 \\ m21 & m22 & m23 & m24 \\ m31 & m32 & m33 & m34 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \rho_4 \\ \rho_4 \\ \rho_4 \end{bmatrix}$$

$$M = \begin{bmatrix} \rho_1 & 0 & 0 & \rho_4 \\ 0 & \rho_2 & 0 & \rho_4 \\ 0 & 0 & \rho_3 & \rho_4 \end{bmatrix}$$

$$\begin{aligned} P_1 &= [1,0,0,0]^T \\ P_2 &= [0,1,0,0]^T \\ P_3 &= [0,0,1,0]^T \\ P_4 &= [0,0,0,1]^T \\ P_5 &= [1,1,1,1]^T \end{aligned}$$

$$\begin{aligned} p_1 &= [1,0,0]^T \\ p_2 &= [0,1,0]^T \\ p_3 &= [0,0,1]^T \\ p_4 &= [1,1,1]^T \\ p_5 &= [\alpha,\beta,\gamma]^T \end{aligned}$$

$$MP_5 = \begin{bmatrix} \rho_1 & 0 & 0 & \rho_4 \\ 0 & \rho_2 & 0 & \rho_4 \\ 0 & 0 & \rho_3 & \rho_4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \rho_5\alpha \\ \rho_5\beta \\ \rho_5\gamma \end{bmatrix}$$

$$\begin{aligned} \rho_1 &= \rho_5\alpha - \rho_4 \\ \rho_2 &= \rho_5\beta - \rho_4 \\ \rho_3 &= \rho_5\gamma - \rho_4 \end{aligned}$$

$$M = \begin{bmatrix} \rho_5\alpha - \rho_4 & 0 & 0 & \rho_4 \\ 0 & \rho_5\beta - \rho_4 & 0 & \rho_4 \\ 0 & 0 & \rho_5\gamma - \rho_4 & \rho_4 \end{bmatrix}$$

Since M is up to a scale:

$$M = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix}$$

To find x, consider the projection of the center of projection:

M projects every point in P3 to P2, with the exception of the center of projection O.

M is rank 3. So O is the non trivial null space of M.

$$M.O = 0$$

$$MO = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix} \begin{bmatrix} O_x \\ O_y \\ O_z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Solving for O:

$$O_x = \frac{1}{1 - \alpha x}$$

$$O_y = \frac{1}{1 - \beta x}$$

$$O_z = \frac{1}{1 - \gamma x}$$

Repeat all the above steps for the second camera …

$$M' = \begin{bmatrix} \alpha' x' - 1 & 0 & 0 & 1 \\ 0 & \beta' x' - 1 & 0 & 1 \\ 0 & 0 & \gamma' x' - 1 & 1 \end{bmatrix}$$

$$\begin{aligned} O'_x &= \frac{1}{1 - \alpha' x'} \\ O'_y &= \frac{1}{1 - \beta' x'} \\ O'_z &= \frac{1}{1 - \gamma' x'} \end{aligned}$$

Now let's use the epipoles (remember that the epipoles are the images of the center of projections of the other camera) to completely find M and M'.

$$\begin{aligned} MO' &= \sigma e \\ M'O &= \sigma' e' \end{aligned}$$

$$\sigma \neq 0 \quad \sigma' \neq 0$$

$$MO' = \sigma e$$

$$M = \begin{bmatrix} \alpha x - 1 & 0 & 0 & 1 \\ 0 & \beta x - 1 & 0 & 1 \\ 0 & 0 & \gamma x - 1 & 1 \end{bmatrix}$$

$$\begin{aligned} O'_x &= \frac{1}{1 - \alpha' x'} \\ O'_y &= \frac{1}{1 - \beta' x'} \\ O'_z &= \frac{1}{1 - \gamma' x'} \end{aligned}$$

$$\begin{aligned} \alpha x - 1 + 1 - \alpha' x' &= \sigma(1 - \alpha' x')e_x \\ \beta x - 1 + 1 - \beta' x' &= \sigma(1 - \beta' x')e_y \\ \gamma x - 1 + 1 - \gamma' x' &= \sigma(1 - \gamma' x')e_z \end{aligned}$$

$$
\begin{aligned}
\alpha x - \cancel{1} + \cancel{1} - \alpha' x' &= \sigma(1 - \alpha' x')e_x \\
\beta x - \cancel{1} + \cancel{1} - \beta' x' &= \sigma(1 - \beta' x')e_y \\
\gamma x - \cancel{1} + \cancel{1} - \gamma' x' &= \sigma(1 - \gamma' x')e_z
\end{aligned}
$$

$$
\begin{bmatrix} \alpha & -\alpha' & \alpha' e_x \\ \beta & -\beta' & \beta' e_y \\ \gamma & -\gamma' & \gamma' e_z \end{bmatrix}
\begin{bmatrix} x \\ x' \\ \sigma x' \end{bmatrix}
=
\begin{bmatrix} \sigma e_x \\ \sigma e_y \\ \sigma e_z \end{bmatrix}
$$

Solve for $\sigma x'$

$$\begin{bmatrix} \alpha & -\alpha' & \alpha' e_x \\ \beta & -\beta' & \beta' e_y \\ \gamma & -\gamma' & \gamma' e_z \end{bmatrix} \begin{bmatrix} x \\ x' \\ \sigma x' \end{bmatrix} = \begin{bmatrix} \sigma e_x \\ \sigma e_y \\ \sigma e_z \end{bmatrix}$$

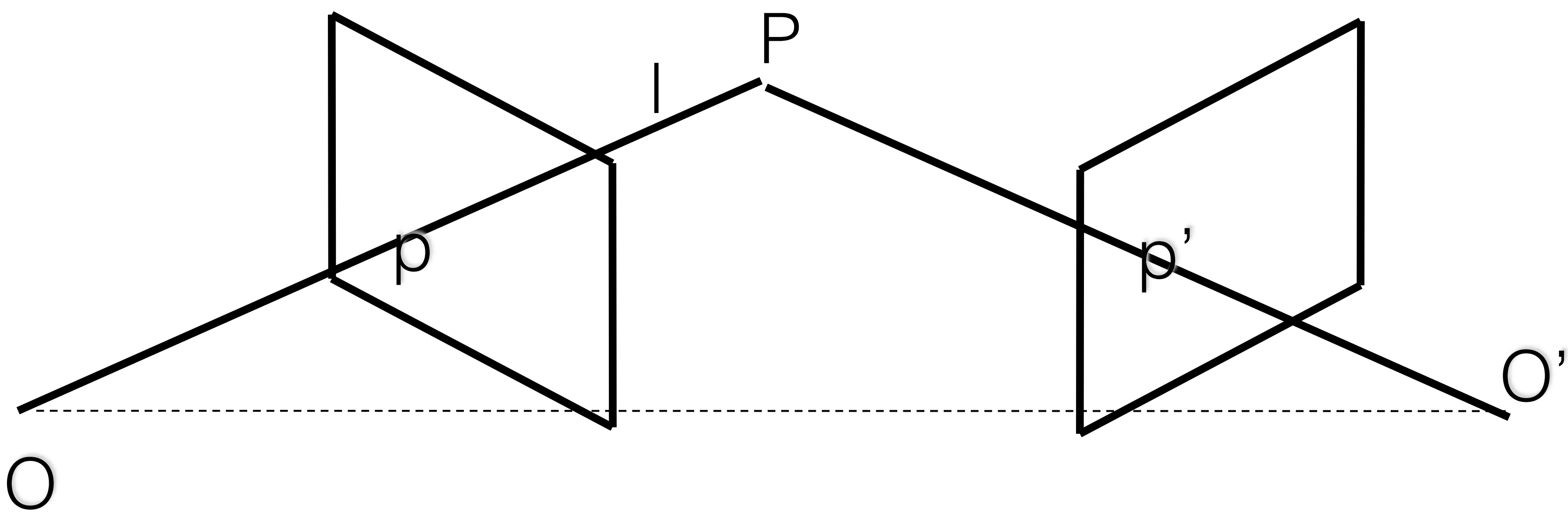
$$\begin{bmatrix} p_5 & p_5' & v \end{bmatrix} \begin{bmatrix} x \\ x' \\ \sigma x' \end{bmatrix} = \sigma e$$

$$\cancel{\phi} x' = \cancel{\phi} \frac{e^T (p_5 \times p_5')}{v^T (p_5 \times p_5')}$$

And similarly for x:

$$x = \frac{e'^T(p_5 \times p_5')}{v'^T(p_5 \times p_5')}$$

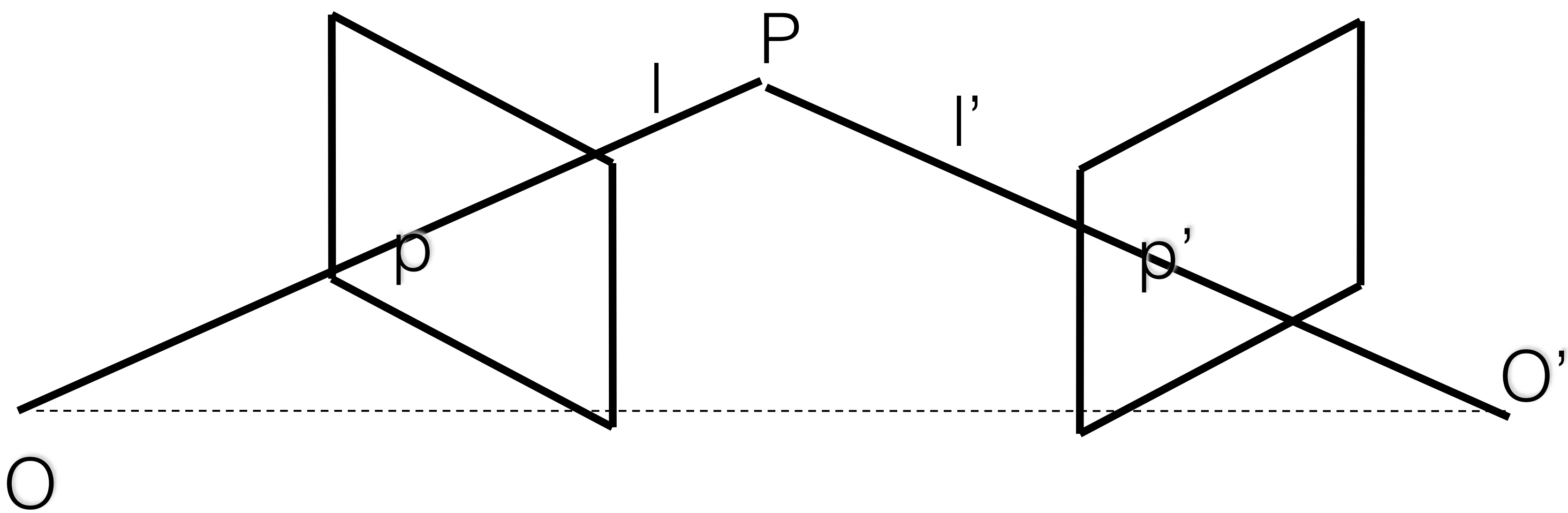
Now we have M and M', O and O', so we can reconstruct any point in P3 given its corresponding images p and p'.



$$\lambda O + \mu[O_x p_x, O_y p_y, O_z p_z, 0]^T$$

Line l that passes through O and p

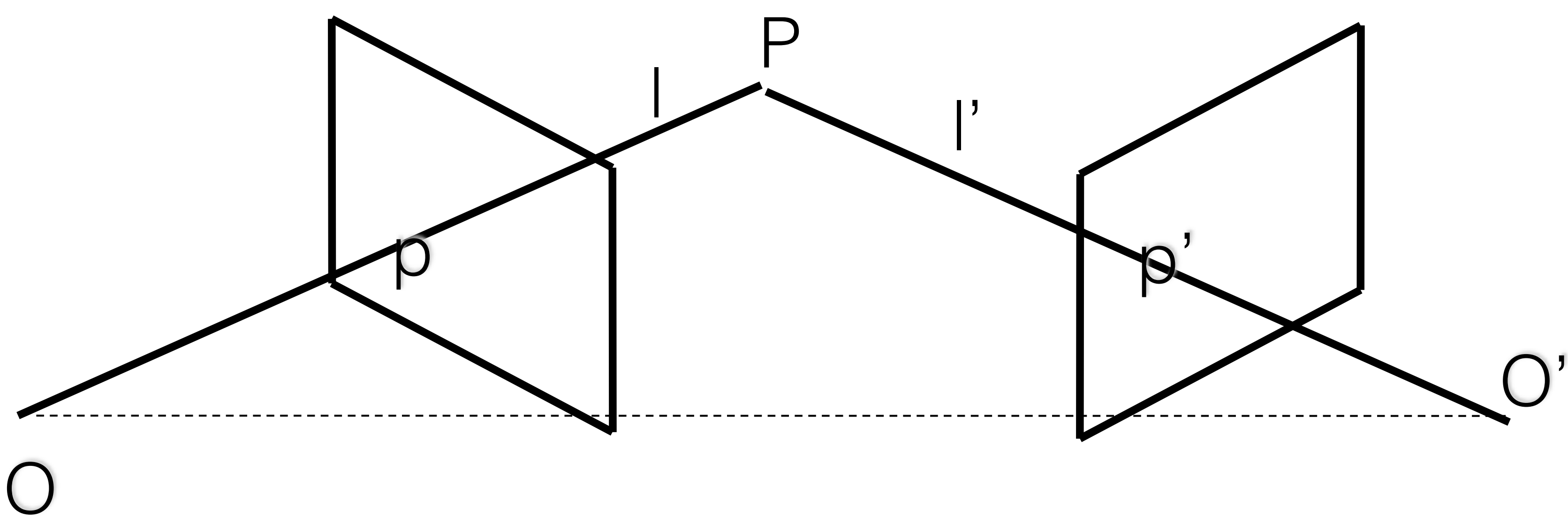
Similarly, for p'



$$\lambda' O' + \mu' [O'_x p'_x, O'_y p'_y, O'_z p'_z, 0]^T$$

Line l' that passes through O' and p'

P lies in the intersection of l and l’



$$\lambda O + \mu[O_x p_x, O_y p_y, O_z p_z, 0]^T = \lambda' O' + \mu'[O'_x p'_x, O'_y p'_y, O'_z p'_z, 0]^T$$

$$\lambda O + \mu[O_x p_x, O_y p_y, O_z p_z, 0]^T = \lambda' O' + \mu' [O'_x p'_x, O'_y p'_y, O'_z p'_z, 0]^T$$

$$\begin{bmatrix} O_x & O_x p_x & -O'_x & -O'_x p'_x \\ O_y & O_x p_y & -O'_y & -O'_y p'_y \\ O_z & O_z p_z & -O'_z & -O'_z p'_z \\ 1 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \\ \lambda' \\ \mu' \end{bmatrix} = 0$$

Compute its SVD: UDV'.
The solution is given by the column of V corresponding to the smallest ev.